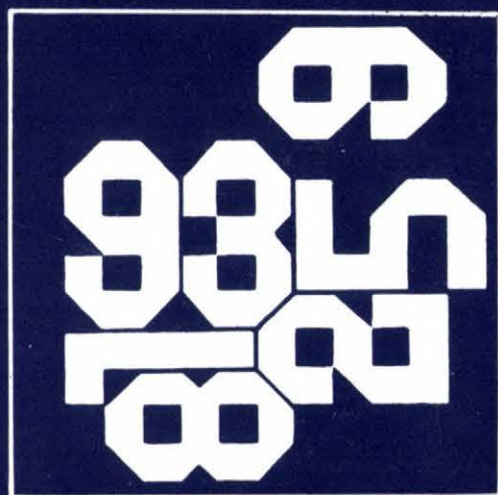


MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest



Magyar Tudományos Akadémia
Számítástechnikai és Automatizálási Kutató Intézete

GYÁRTÓRENDSZEREK FUNKCIONÁLIS ANALÍZISE
ÉS SZINTÉZISE

BERNUS PÉTER

Tanulmányok 189/1986

A kiadásért felelős:

DR. KEVICZKY LÁSZLÓ

Főosztályvezető:

NEMES LÁSZLÓ

*Jelen tanulmány eredetileg a szerző
kandidátusi disszertációja*

ISBN 963 311 218 4

ISSN 0324-2951

TARTALOMJEGYZÉK

old.

1. BEVEZETÉS: A gépipari integrált anyag és adatfeldolgozó rendszerek innovációs folyamata	1
1.1. Eszközök és módszerek fejlesztése	4
1.2. A megvalósulási folyamat közege	6
1.3. Célok kitűzése	7
1.4. Funkcionális tervezés	8
1.5. Műszaki tervezés	11
1.6. Kivitelezés, üzemeltetés	23
1.7. Az innovációs folyamat tárgyalásának összefoglalása	26
2. A FUNKCIONÁLIS TERVEZÉS IRODALMÁNAK ÁTTEKINTÉSE	30
2.1. Strukturált analízis és tervezési módszer /SADT/	33
2.2. A Petri háló alapu modellezési technikák	51
2.3. Rendszerleíró adatbáziskezelő rendszerek	70
3. CÉLOK ÉS MUNKAMÓDSZER	81
3.1. A funkcionális tervezés módszereire irányuló kutatási koncepció	81
3.2. Stratégia és munkamódszer a rendszertervezői rendszer egyes elemeinek megvalósítására	87
3.3. További lehetőségek	94
4. SAJÁT EREDMÉNYEK A FUNKCIONÁLIS RENDSZERTERVEZÉS TERÜLETÉN	97
4.1. Funkcionális modellek adatbázisának definiálása és kialakítása	97
4.1.1. A funkcionális modellezés módszerei közötti választás indoklása	99
4.1.2. A SADT és az ISDOS rendszer összekapcsolása	103

II.

4.1.3. A választott módszerek és kifejlesztett eszközök gyakorlati alkalmasságának bizonyítása, megoldásra váró problémák föltárása	114
4.1.4. A funkcionális modellek továbbfejlesztésének néhány eredménye	124
4.2. A funkcionális notáció és adatbázis kapcsolata	137
4.2.1. A funkcionális notáció formalizálása, definíciók és alapfogalmak	138
4.2.2. A funkcionális architektúra szemantikája	160
4.2.3. A FA teljessége és ellentmondásmentessége	170
4.2.4. Eszközüválasztás és kísérletek az SDLA rendszerrel	177
5. ÖSSZEFOGLALÁS	184
IRODALMI HIVATKOZÁSOK	187
I. FÜGGELÉK: Asszociatív adatbázisfelület bináris relációk kezelésére	195
II. FÜGGELÉK: SADT diagrammok analizise SDLA segítségével	199

A j á n l á s

F E L E S É G E M N E K

Ennek a fejezetnek az a célja, hogy olvasójával megismertesse a dolgozat tárgyát választott funkcionális tervezési módszerek helyét és szerepét a gépipari gyártórendszerek életében. Ezt a célt úgy kívánja elérni, hogy a rendszerek életfolyamatát, annak specifikumait tárgyalja, mindig rámutatva a funkcionális tervezési módszerek abbeli helyére és jelentőségére.

1. BEVEZETÉS: A gépipari integrált anyag és adatfeldolgozó rendszerek innovációs folyamata

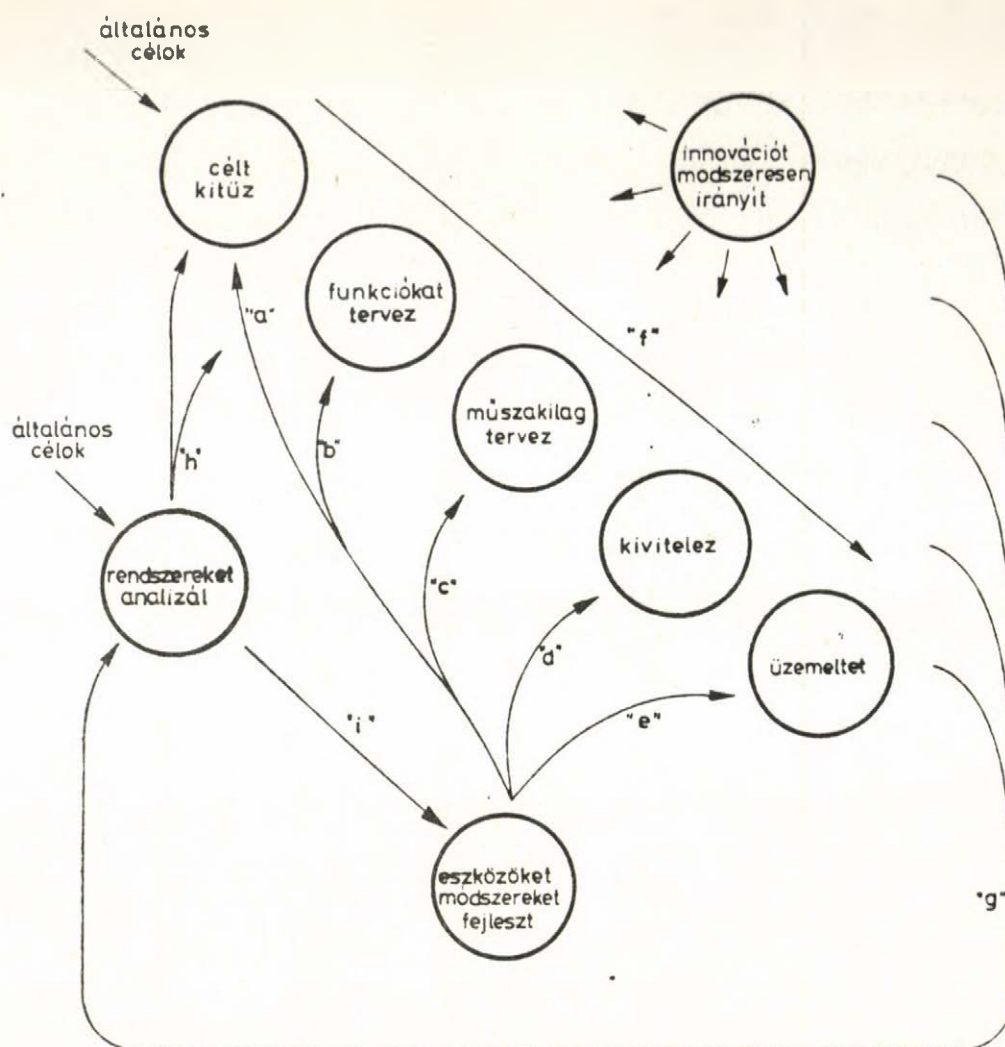
Bármely nagyobb rendszer élete felosztható önmagában is vizsgálható szakaszokra. Ezek a szakaszok egy-egy szakterület tevékenységeit tömörítik. Az ötlet kialakulásától a tervezésen át a megvalósulásig és használatig a tevékenységek nem ténylegesen sorbakapcsolva jelentkeznek, hanem funkcionális sémába rendezhetők. Sokan és sok szempontból vizsgálták már ezt a folyamatot. Minél nagyobb rendszer életéről van szó, annál nagyobb mennyiségű és szövevényesebb információ gyűrűzik az idők folyamán vele együtt, így érdekünkben áll azt átgondolni, vajon mi szabja meg ezen információ tovaterjedését emberek, számítógépes adattárak és papírra írt dokumentumok között. Az információáramlás támogatása számítógéppel csak ennek az ismeretnek a birtokában lehet reményt keltő, sikeres vállalkozás.

Vizsgálatunk tárgya nem akármely nagy rendszer, minket jelenleg a gépipar gyártórendszerei és gyártócellái érdekelnek, bár kétségtelen, hogy nem egy más szakterület problémái hasonlóak lehetnek. A szakosodást azért is meg kell tenni ezen a ponton, mert kétszeresen összetett probléma előtt állunk: nemcsak a megvalósulási folyamat (jobb híján ezt a szót magyarítottam innováció helyett) egyes lépcsői miatt mozgunk több szakmát átölelő közegben, hanem maguk a vizsgált rendszerek is interdiszciplináris alapokon nyugszanak.

Nem tekinthetünk el az egyedi rendszerek és a fejlődés összefüggéseinek számbavételétől sem, mert a gyártórendszerek életciklusa eddigi tapasztalataink szerint olyan hosszú, mely idő alatt az egyes diszciplínákon belül is számottevő a változás gyakorisága.

A megvalósulási folyamat rövidítése számos egyéb jótékony hatása mellett a fejlődés folyamatosságának irányába hat és várható, hogy a fejlődési holtvágányok számát, összráfordítását szintén csökkenti [BER82a].

Az 1.1 ábra sematikusán mutatja be a teljes folyamatot, úgy, mint egymással összefüggő feladattételek sorát. Ezek a feladatok természetesen több lépésben is tovább bonthatóak. Az "f"



- 1.1 ábra -

A megvalósulási folyamat

nyil azt mutatja, hogy milyen egy adott rendszer esetén az információ fő áramlási iránya. Az "a"-tól "e"-ig terjedő nyilak pedig azt mutatják, hogy minden feladathoz eszközökre és módszerekre van szükség. Az "i" és "h" nyil végezetül az egyedi rendszerek életéből levont tapasztalatokat és kényszereket szimbolizálja - melyeknek forrása a létező és elgondolt rendszerek általános rendszeranalízise. A "h" nyil olyan kényszereket jelent, melyek a meglévő rendszerek ismeretében korlátozzák az új rendszer célkitűzéseit (u.m. piaci versenyképesség feltételei, funkcionális elvárások - okulva a létező rendszerek tapasztalataiból - ajánlások, szabványok, konzervtervek és tervrészletek: általában minden olyan információ, mely több rendszer ismeretéből levont következtetésből fakad).

1.1 Eszközök és módszerek fejlesztése

A rendszerek analízise az eszközök és módszerek fejlesztése számára igényeket hoz létre. Ez természetesen nem zárja ki az öntörvényű fejlődést sem, azonban az öntörvényűség a rendszer-szintézis ellen ható folyamat, amelynek legalább egy rendező-elve van szüksége ahhoz, hogy a rendszerek fejlődésével ne interferáljon. Mindennek tükröződnie kell a rendszeranalízisben és az eszközök, módszerek fejlesztésében is. A rendszeranalízis egyik feladata, hogy a meglévő rendszerek tapasztalatainak

és az általános céloknak az ismeretében teljes, vagy részleges követelményrendszert állítson föl az összes, vagy egy kiválasztott diszciplína számára. Ez a feladat nem végezhető el csupán egyetlen szakterület ismeretében.

A rendszeranalízis módszertana eszközeit és végrehajtását tekintve rokon a konkrét rendszerek kidolgozásával ("f" folyamat), azonban célja különbözik attól. Az eszközök és módszerek (software és hardware objektumok, technológiák, szervezési és végrehajtási módszerek...) fejlesztése diszciplínákon belül folyik, részben a rendszerek analízise által létrehozott "i" követelmények hatására, részben önindíttatásra, a követelményeket mintegy előre vetítve, extrapolálva. Mivel nem várható, hogy minden diszciplína fejlődése a rendszerek fejlődésével szinkronban menjen végbe, ezért csak a tiszta funkciókat megvalósító primitívek kutatása és fejlesztése képes a két fejlődési folyamat konvergenciáját biztosítani.

Az evolúció lehetősége önmagában nem garancia, a revolúciót nem helyettesíti, sőt a tiszta funkciókat megvalósító primitívek és a hozzájuk kapcsolódó szakterületi elméletek egyben a revolúció objektív feltételei is. A módszerek fejlesztéséről ugyanez mondható el, azonban az ábrán vázolt rendszer manapság igen erős tranziens állapotban van: a diszciplínák közötti

viSSZacsatolások még csak kialakulóban vannak. Így jelenleg hiányzik egy olyan általános "i" követelményrendszer, amely az eszköz és módszerfejlesztőket a gyártórendszerek és gyártócellák szükségleteinek irányába terelné.

Ez a követelményrendszer természetesen nem lehet független a gyártórendszerek és cellák időtálló koncepciójától, Az elemzésnek nemcsak a műszaki, hanem a társadalmi és gazdasági környezetre, a környezettel való kölcsönös viszonyra is ki kellene térnie.

1.2 A megvalósulási folyamat közege

Az "f" folyamat közege a fővállalkozó. Eszköze a rendszertech-nika, mely az a,b,c,d... módszereket használja - esetleg a bonyolult tervezési és mamagement információs folyamatot számítógéppel is támogatva. A folyamat automatizálásának fokozatai vannak. A legegyszerűbb szint, ha a mamagement csoportot ellátjuk egy saját nyilvántartási rendszerrel, mely nem kapcsolódik a folyamathoz automatikusan, csak ezen a csoporton keresztül. A központi adatgyűjtés megemeli az innovációs folyamat irányításának szintjét, mert az információ forrását függetleníti az irányító tevékenységtől. A központi adatgyűjtés automatizálása a megvalósulási folyamat számítógépes támo-

gatásának függvénye. Itt a vezetési információhoz a folyamat melléktermékeként in statu nascendi lehet hozzáférni. Az információ előkészítése sok esetben szükségessé teszi, hogy szubjektív ítéletből származó, vagy hiányos információkat is exakt módon kezeljünk.

1.3 Célok kitűzése

A megvalósulási folyamat első szakasza a célok kitűzése. Ez igen gazdag tevékenységi kört fed, mely a fejlesztési politika kialakításától a termékkel szemben támasztott követelmények megfogalmazásáig terjed. Ennek a részfolyamatnak a végrehajtói különböznek a leginkább egymástól, mivel általában nemcsak szakterületük, hanem beosztásuk is erősen eltérő. További nehézség ebben a szakaszban, hogy a közbelső eredmények nem szükségszerűen jelennek meg explicit formában, részint mert esetleg az egyes szakaszok egy és ugyanazon személy fejében mennek végbe, részint a megszokás miatt szájhagyomány útján terjednek. A helyzeten egyedül szervezési megoldásokkal nem lehet segíteni, mivel csupán a szükséges információ megszerzése nem elegendő, azt sokféle szempontból meg kell tudnunk jeleníteni (duplikáció, szerkesztés), ami automatizálás nélkül még az implícit információkon alapuló rendszernél is kevésbé hatékony megoldás. Csak így várható, hogy az explicit informá-

ció valóban megszületik és eljut az érdekeltekhez. A döntéslökészítő embercsoporttól másként nem várhatunk hatékony, módszeres munkát, csak ha megszüntetjük azt a krónikus kapacitáshiányt, amely az explicit információ meg nem születésének oka. Sok vezető rabja olyan papírgyártásnak, amelyet a rossz szervezet kényszerít rá, nem pedig az elvégzendő munka szükségyszerűsége. Így nem csoda, hogy fontos dolgok (döntések és elképzelések) leíratlanul maradnak - hiszen ennek kényszere igen nagy részben belső és nem külső, miáltal könnyebben elhagyható.

A döntéskoncentráció igényét nem szabad ebből az összefüggésből kiragadni, mert nem az a cél, hogy minden jelenleg is meghozott döntést egy helyre tömörítsünk, hanem az, hogy a jelenleg is meghozott döntéseket ezentúl a megfelelő helyen hozzuk (ma ez decentralizációt követel). Ezért cserébe egy sor olyan döntés meghozatalára van szükség (és nyílik meg a lehetőség), melyet ma a kérdések explicitté tétele és előkészített információ híján nem hoz meg senki.

1.4 Funkcionális tervezés

Az innovációs folyamat egyik mostohagyereke volt, sokhelyütt ma is az a funkcionális tervezés, mely a rendszer feladatát a

maga teljességében meghatározza. Egyszerű termék esetén ennek hiánya nem tűnik szembe első látásra, kimutatása azonban pl. értékelemzés segítségével lehetséges. Bonyolultabb esetekben – mint a gépipari integrált anyag és adatfeldolgozó rendszereké – a rendszermeretek mennyiségi mutatói minőségi változást eredményeznek a megvalósulás folyamatában, mely a funkcionális tervezés hiánya miatt esetleg nem is lesz konvergens.

Székebb-tágabb szakterületeken ki is alakultak modellezési technikák és módszertanok, velük együtt számítógépes eszközök a funkcionális tervezés szakaszára.

A modellezési technikák között vannak időfüggő és időfüggetlen, kötött elemkészlettel vagy anélkül dolgozóak. A módszertanok két koncepció egyikéhez tartoznak: az egyik szerint elő kell írni a tervezőnek, hogy milyen lépéseken keresztül kell az eredményt elérnie, a másik csak eszközöket ad a tervező kezébe (bár igen általánosakat) és ráhagyja, hogy segítségükkel a saját módszere szerint dolgozzon [BIE78]. Mindkét módszernek van előnye is, hátránya is. Valószínűleg kisebb félreértés is az oka annak, hogy a két módszert a módszertan-kutatók így különböztetik meg, hiszen nyilvánvalóan az indirekt módszertanok sem nélkülözhetik a módszeres munkát, tehát kell lennie valamilyen szervezett tevékenységsorozatnak, me-

lyet a tervező követ. A direkt módszertanok csak abban különböznek (esetleg) az indirekttől, hogy a módszer elő van írva, esetleg számítógépes támogatása is az adott munkamódszerre épül. Egy jól felépített direkt módszertan azonban szigorúan meg kell különböztesse saját magát és eszközeit, különben fejlődésképtelen és csak idő kérdése, mikor válik retrográddá.

A tervezőnek ezért olyan tervezőrendszer kell, mely mindenféle modellezési eszközt tartalmaz, lehetőséget ad azok kombinált használatára, és a munkamódszer bizonyos lépéseit automatizálni lehet segítségével. Igen erős érdek fűződik ezért az ismert módszerek és modellezési eszközök összehasonlításához és összekapcsolásához.

A funkcionális tervezés szakaszában részletes és ellentmondásmentes specifikációkat kell létrehozni a tervezett rendszer egyes részeit megvalósító szakterületek számára. A specifikációknak logikailag ellentmondásmentesnek kell lennie, tehát minden funkció ki- és bemenetének egy és csakis egy definíciója lehet és kell is hogy legyen. Világosan rögzíteni kell a funkciók közötti rész-egész és kapcsolati viszonyt, beleértve a funkciók által használt dolgok hasonló viszonyait is. Úgy tűnhet, hogy a funkcionális tervezés lényegének vázlata egy

teljesen determinisztikus folyamat látképe. Ez távolról sincs így! A diszciplínák által definiált funkciófogalmak sokféle fizikai referenciával rendelkezhetnek - esetleg több szakterületen is. A funkcionális tervezés információforrása a szakemberek csoportja lévén, a módszertanokban szükségszerűen ki kell térni ezen embercsoport mozgásformáira - a vezetéstől a kreativitást elősegítő módszereken át az együttműködés feltételeinek megteremtéséig [LAD81],[LAD82].

Külön említést érdemel az értékelemző szemlélet érvényesítése a műszaki tervezés szakaszában. Az értékelemző munka különösen fontos, ha általános, többször is felhasználandó eszközök tervezése a feladat. A flexibilis gyártórendszerek és a technológiai gyártócellák a gépipari termelésben éppen ilyen általános célú gyártókapacitást jelentenek, értékelemzésük ezért is kívánatos.

1.5 Műszaki tervezés

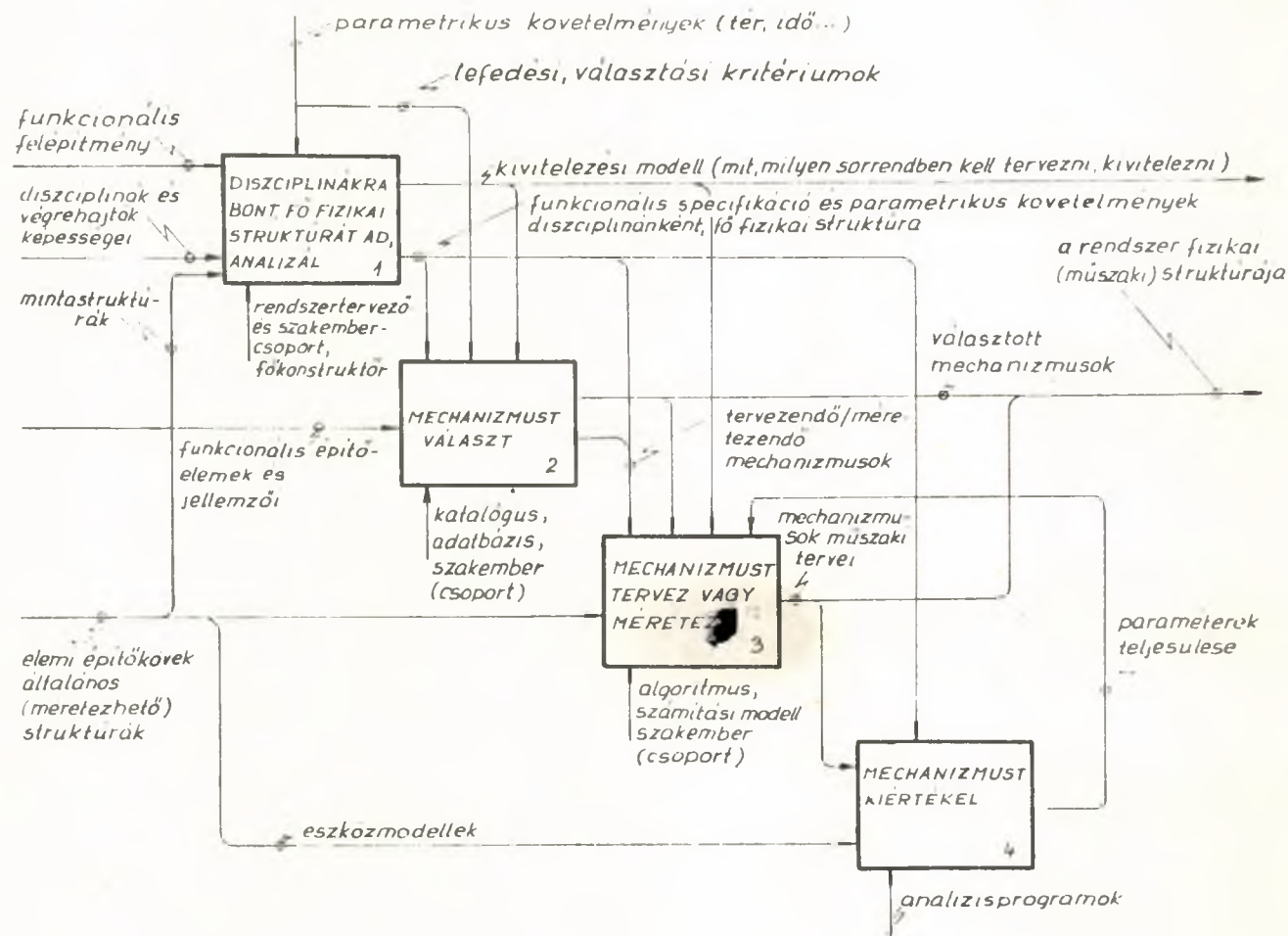
A következő innovációs láncszem a műszaki tervezés. Ennek erősen termékfüggő volta ellenére vannak állandó elemei. Az előző fázisból nincs hirtelen átmenet ebbe a szakaszba, mivel a mechanizmusok hozzárendelésének idején esetenként mennyiségi döntések is közrejátszanak.

1.5.1 A műszaki tervezés modellje

A műszaki tervezés területe igencsak széles, bele kell értenünk a gépészeti egységek (megmunkálógépek, szállító, tároló és egyéb különleges elemek), irányító elektronikus és villamos alrendszerek (számítógépek, általános adatkezelő és folyamat-perifériák, vezérlőegységek), programrendszerek (operációs rendszerek, általános célú software eszközök, vezérlőprogramok) és szervezeti egységek (munkakörök, működési szabályok) tervezését. Az 1.2 ábra a műszaki tervezés modelljét tárja az olvasó elé, mégpedig olyan módon, hogy abban a több szakterületet átölelő tevékenységek és azok különféle végrehajtási módszerei is képviselve legyenek. Az irodalom kategóriákba sorolja a tervezési folyamat módszereit aszerint, hogy hogyan jutunk el a specifikációtól az azt megvalósító fizikai struktúrához. Egy-egy szakterület szemszögéből vizsgálva ez meg is felel a követelményeknek, bár már ott is fölmerül a kérdés: mi módon döntjük el egy adott feladat esetén, hogy melyik diszciplína alkalmazása lesz a célszerű. Erősen összetett, heterogén rendszerek esetén a döntés még előrébb tolódik.

1.5.1.1 Diszciplínákra bontás

A diszciplínák képességeinek és a rendszerrel szemben támasz-



A műszaki tervezés modellje

1.2. ábra

Tervezte	Bernus	Cím	A műszaki tervezés modellje				Cég	SZTAKI
Rajzolta		1.2. ábra				A projekt megnevezése		
Gépelte								
Ellenőrizte								
Készítés	1982 június	Munkapéldány		Oldalszám	A szülő változat száma	Változat szám		
Az elavulás időpontja	Jóváhagyva	Aktuális példány	1		A szülő változatszám	Változatszám	1	

tott parametrikus követelményeknek alapján nagy vonalakban fizikai struktúrát kell adni a tervezett rendszernek. Ennek a részletei már egy-egy szakterület felségterületére esvén annak a területnek a módszereivel megtervezhetők. Ez az első bontás természetesen szervezési és analízisbeli kötöttségekkel is jár. A területek csatlakozó felületeit vagy előre határozzuk meg, vagy pedig szervezéssel gondoskodunk arról, hogy e tekintetben a tervezés előrehaladtával megállapodások szülessenek. (Erre a feladatra alkalmas a tervezési és kivitelezési folyamat modellje.)

A teljes rendszerrel szemben támasztott parametrikus követelményeket is le kell bontani a tervezendő mechanizmusok szintjére. Ez a lebontás a paraméterek típusától függő analízismódszerekkel ellenőrizhető. Mivel ez a tervezési lépés nem tartozik egyetlen szakterületre sem, nem meglepő, hogy módszerei és eszközei a további három lépéstől ma még elmaradnak. Szorosan csatlakozik ez a tevékenység a funkcionális tervezéshez: a megvalósulás e szakasza a funkcionális tervezési szakaszból táplálkozik, emellett a végrehajtásához szükséges képzettség is hasonló. A funkcionális tervezés eszközei mellett ez a fázis a szimulációra és a szervezéstechnikára épül.

Ma a műszaki tervezés e részfolyamata teljes mértékben szub-

jektív alapokon nyugszik, és annál sikeresebben oldható meg, minél átfogóbb szakmák fölötti ismeretekkel rendelkezik a tervezést és megvalósítást vezető főkonstruktor. Nemcsak a szakmák általában ismeretes képességeit kell itt mérlegre tenni, hanem a rendelkezésre álló tervezői, kivitelezői kapacitást és minőségét is.

1.5.1.2 Mechanizmusválasztás és tervezés

A folyamat nem időben, hanem funkciójában különül el a műszaki tervezésen belül. A mechanizmusok kiválasztása és tervezése során finomodik a fizikai struktúra, míg nem előáll a kész műszaki terv. A finomítás során újra meg újra ellenőrizhető, hogy a terv milyen mértékben teljesíti a követelményrendszerben foglaltakat (mennyire konform a funkcionális modellel), és ez az információ közvetlenül felhasználható a terv további alakításának vezérlésére. Ilyen evolúciós természetű eljárást próbál pl. a gépészeti konstrukció automatizálásában meghonosítani Yoshikawa paradigma modellje [YOS81].

1.5.2 A műszaki tervezés néhány specifikus területe

1.5.2.1 Időbeli és térbeli modellezés. A mennyiségi döntések meghozatalának egyik eszköze az időbeli és térbeli modellezés,

melynek számos (rajzos, szimulációs és tényleges fizikai) modellezési technikája ismert. Ezek a módszerek lehetnek determinisztikusak és sztochasztikusak, ezen belül diszkrét és folytonosak is. Bármely szimulációs modellt tekintsük is, megkonstruálásának első lépése mindenképpen a funkcionális modell megalkotása az adott szimulációs nyelven. Ha tehát mind a funkcionális mind a szimulációs modelleket számítógépen készítjük, önként adódik ezek közös részének automatikus összevetése, ellentmondásainak kiszűrése, vagy az egyik átfordítása a másikra. A szimulációs modelleket fől lehet használni arra, hogy valamilyen időbeli vagy egyéb mennyiségi követelmény kielégíthetőségét ellenőrizzük. A térbeli modellek lehetnek topológiaiak, felületmodellek és volumetrikus (test-) modellek.

A gyártórendszerek telepítésének tervezése sok közös vonást mutat az építészeti tervezéssel, ugyanis a telepítés terve nem csupán geometriai modellek szabta korlátok között készül, hanem sokféle egyéb követelmény kölcsönhatására tekintettel. A szállítási utak és tárhelyek topológiája és mérete alapvető hatással van a rendszer időbeli működésére. Az időbeli vizsgálat számos paraméterét, a szimulációs modell struktúrájának egy részét ezek a térbeli modellek szabják meg.

1.5.2.2 Programozás

A programszintézis módszertanai, vagy az azt segítő rendszerek mind foglalkoznak valamely formában a programok definiálásának, kódolásának kérdésével, és a programot létrehozó ember (embercsoport, vagy ember-gép együttes) szerepével. A software-ház megközelítés általában egy specifikációs nyelven alapul (pl. [YOU75], [WAR74], [JAC75], [STA76], [WEG72], [SZE79]). Ezen a nyelven le lehet írni a rendszer funkcióit és adatainak szerkezetét. A tiszta funkcionális tervezési módszerekhez képest itt az az eltérés, hogy kötött metaelemkészletet használnak, elsősorban a struktúrált programozás primitívjeire alapulva -- kibővítve azokat a mai modern programozási nyelvek (Algol 68, PASCAL, C, MODULA, PEARL, MARY, GESAL, ADA...) adat-, programmodul- és esetleg eseménykategóriáival. A modellezési módszer megválasztása mellett az embercsoport munkájára is vannak a gyakorlatban sikerrel alkalmazott eljárások, melyek alapja, hogy erősen dokumentációorientáltak (döntések explicitálása) és az emberi funkciók ellenőrzöttek (design walkthrough, egoless programming...). Mindamellet a szakterület emberének ezeket a formákat tartalommal kell megtöltenie, olyan magasabbszintű eszközök birtokában mint pl. az erőforrásmegosztás módjai (és kifejezési formája az adott modellezési módszerrel) vagy a rekurzív fabejárási algoritmu-

sok stb. A magasabb szintű eszközök, meghonosodott absztrakciók hiányában csak gyatra tervek születhetnek, ami a programrendszer főlegesen sajátsgaiban, karbantarthatatlanságában jelentkezik (nem módosítható, nem érthető).

A szintézismódszerek ellentettjére, a programstruktúrák automatikus analízisére, programok szimbólikus végrehajtására is van példa. (Ez elvi jelentőségű módszer, mert a pontonkénti ellenőrzés helyett teszt-trajektóriák vizsgálatára ad módot.)

A mesterséges intelligencia módszereinek itt is bő alkalmazási területe van. A terület fejlődésének eredményeire -- ellentétben a "software-ház" módszerrel -- a számítógéppel segített evolúciós programtervezés lehetne a megfelelő megnevezés. Alapvető eltérés az előzőektől, hogy szakértői rendszert feltételez (automatikus programanalízis, magasabb szintű építőelemek fölismerése és alkalmazása, programmódosítási képességek).

1.5.2.3 Gépészeti konstrukció

A gépészeti konstrukciós tervezés erősen konzervatív alapokon nyugszik, mivel a konstrukciós változtatások hatásai a mai eszközökkel jórészt csak kísérletileg ellenőrizhetők. A szu-

perszámítógépek megjelenése várhatóan radikálisan fog változtatni ezen a helyzeten. Vannak már ma is olyan gépészeti területek, melyeken jelentős eredmények születtek direkt tervezési eljárások alkalmazásával (pl. turbinatervezés). Mindez újfajta kapcsolatok alapja a diszkrét gépipari gyártórendszerek megvalósulási folyamatában. Az alapvető irányzat a megvalósulási folyamat struktúrájának egyszerűsödése (ezzel párhuzamosan az innováció relatív rövidülése), ugyanis a direkt eljárások a szükséges visszacsatolások számát csökkentik. Ugyanekkor a tervezéshez és a megvalósításhoz szükséges eszközök válnak egyre összetettebbé, általánosabb célúvá. Így tükröződik szűkebb területünkön az az általános megállapítás, hogy a jelenkor műszaki forradalma nem konstrukciós, hanem technológiai eredetű, a konstrukciókban csak megtestesül, de nem azokból származik. Valószínű, hogy a bevezető gondolatsor problémája az eszközök és rendszerek fejlődésének szinkronitásával kapcsolatban időlegesen hol így - hol úgy vetődik föl: egyszer a technológia fut előre (amikor is a konstrukció kap szárnyakat, és a technológia konzerválódik), majd újabb technológiai forradalom következik, mellyel a konstrukciónak kell megpróbálnia fölvenni a lépést.

A manipuláció rugalmasságával szemben támasztott igények növekedésével (akadálykikerülés, valós időben számított optimális

trajektóriák, alak- és szituációfelismerés) nagysebességű, esetleg párhuzamos számításokra alkalmas eszközökre van szükség. A robotok tervezése (dinamikus merevség, vezérlőalgoritmusok vizsgálata...) önmagában is igényli a funkcionális tervezési módszereket, a geometriai modellezési eljárásokat, a statikai és dinamikai modelleket, valamint a vezérlés és a gépészeti konstrukció együttes viselkedését leíró dinamikai szimulációs modelleket.

1.5.3 A kivitelezés tervezése

A teljes gyártócella vagy gyártórendszer funkcionális felépítményének megtervezése végén az elsőszámú műszaki tervezési feladat a funkciók szétosztása azokat megvalósító mechanizmusokra. Túl a különféle mennyiségi követelményeken a lehetséges megoldások száma gyakran még így is jelentős. A megvalósítás költségeinek analízise külön módszert igényel, mégpedig olyat, mely képes hiányos információk, becslések és valódi adatok kombinációját figyelembe venni. Ebben a fázisban még több változattal kell számolnunk. Minden változathoz a tervezés és kivitelezés tevékenységeinek funkcionális modellje tartozik, mely a tevékenységek feltérképezésére, időzítésére és eszköz-szükségletének feltárására irányul.

1.5.4 Dokumentáció

A műszaki tervezés eredménye mindenféle rendű és rangú dokumentáció. Az előzőekben taglalt kivitelezési modell nem más, mint az ebben a műszaki adatbázisban előforduló objektumok (műszaki modellek, leírások) funkcionális meghatározása. Ezek viszonyának automatikus kezelése föltételezi a formális modellek, grafikus ábrázolási formák és szöveges leírások összefüggésének ismeretét. Ez a feladat tartalmi szinten csak ma még nem létező intelligens rendszerekkel volna általánosan megoldható, azonban a kivitelezési modell alapján viszonylag egyszerűen nyomonkövethető az információ útja és így egy-egy tartalmi változtatás hatásláncát formálisan is föl lehet göngyöltetni. A dokumentáció kérdésére még vissza kell térnünk a rendszer üzemeltetése és karbantartása kapcsán.

1.5.5 Megbízhatósági kérdések

A megbízhatósági kérdések a megvalósulási folyamaton belül külön nézőpontot testesítenek meg. A funkcionális tervezés - na teljesen eltekint az alkalmazható eszközök ismeretétől - nem mond semmit a rendszer megbízhatóságáról, legfeljebb a megbízhatósági analízishez szükséges egyik kiinduló információt szolgáltatja. A rendszerek felügyeleti, diagnosztikai és

őnjavító funkcióit azonban ez a tervezési szakasz rögzíti. Mivel egy nagy rendszernek elvben minden szintjén szükség lehet felügyeleti és diagnosztikai funkciókra, a funkcionális tervezés során ezek helye meghatározható - függetlenül attól, hogy egyes ilyen funkciókat később, a megbízhatósági analízis eredménye alapján szükséges-e megvalósítanunk.

A megbízhatóság kérdése egyedi, nagy rendszerek esetén a sorozattermékekétől eltérően vetődik föl. Az esetek zömében a rendszerek előállítási technológiája nem eléggé reprodukálható ahhoz, hogy stabil megbízhatósági értékekkel számolhassunk. A rendszerépítés folyamán így az alapkérdés annak megtalálása, hogy mik lesznek a reális hibahelyek, majd olyan funkcionális terv készítése, mely a meghibásodás hatását kivédi, vagy mederbe tereli. Ez a felügyeleti és diagnosztikai funkcióknak az egész rendszer tényleges funkcióival alkotott kombinációját feltételezi. Ma még hiányzik az a gyakorlatilag is alkalmazható módszertan, melynek alkalmazásával a tervezett rendszerekben könnyű a hibakeresés és javítás. (Egy elv például, mely a jól diagnosztizálható és javítható rendszer tervezésénél betartandó, hogy a rendszer funkcionális és fizikai struktúrája legyen megegyező.)

A jelenlegi ismeretek szerint a megbízható rendszerek tervezé-

se legalább egy iterációs lépést igényel: a funkcionális tervezés során hipotetikusan beépítjük a felügyeleti és diagnosztikai funkciókat, a műszaki tervezés eredményének analízisa után pedig áttervezzük a funkcionális felépítményt [KOV82].

A diagnosztikai elvek mellett szükség volna olyan szabványos elvek kidolgozására, melyek a funkcionális tervező kezében útmutatóul szolgálhatnának. (Így például ki lehetne dolgozni jellegzetes struktúrákat a rendszerek generálási funkcióira, tesztelési stratégiákra, a javítási és karbantartási funkciók közötti összefüggések feltárására.)

1.6 Kivitelezés, üzemeltetés

A kivitelezés szervezésének a kivitelezési modell az alapja. A kivitelezési tevékenységek technológiája szabja meg a műszaki dokumentáció formáját. Az innováció folyamatosságának feltétele, hogy a kivitelezés követelményeit és lehetőségeit a műszaki tervezés számára hozzáférhetővé tegyűk. A lehetőségek a tervezésbe bevont technológustól a termékgazdán keresztül a technológiai adatbankok tervezői segédletéig terjednek.

Különös figyelmet érdemel a kivitelezés azon része, ahol

eltérő elvű részrendszereket csatlakoztatunk: ez a rendszerintegrálási tevékenység. A kivitelezés legnagyobb nehézségei megelőzhetőek és az integrálás drága folyamata rövidíthető, ha a kivitelezési modellek tartalmazzák az önálló ellenőrzési feladatokat, s ha azokat ténylegesen végre is hajtják a végleges állapotú részrendszereken. A management eszközei itt a kivitelezési terven és a költségvetési valamint erőforrás modelleken nyugszanak (sokszor csak fejben meglévő modellek). Hangsúlyozni kell, hogy ezeket a módszereket csak akkor remélhetjük sikerre vinni, ha egy bizonyos kritikus színvonal fölé emeljük infrastruktúránkat. Ha ez nem sikerül, akkor az elszigetelt részalkalmazások előnyeit elfedik az előkészítéssel járó energiabefektetések - ami egyértelműen a módszerek kudarcához, sőt további lehetőségeik súlyos visszavetéséhez vezet.

Bonyolult rendszereknél a próbaüzem és a rendszer átadása a felhasználónak éppoly gondosan megtervezendő, mint a rendszer maga. Ember-gép rendszerek esetén már a funkcionális tervezésnél kezdődik a felhasználás módjának és menetének tervezése üzemelési és üzemeltetési modellek, forgatókönyvek készítésével.

A gyártórendszerek és hasonló nagyságrendű ember-gép rendsze-

rek karbantartása is hozzátartozik a megvalósulási folyamat-hoz. Az üzemeltetőnek és a gyártónak egyaránt gondot okozó tevékenység problémája, hogy egy ilyen rendszer karbantartása és javítása többféle szakembert igényel, emellett még az egyes szakemberek sem cserélhetők föl egymás között. A tényleges rendszer és a dokumentáció kapcsolata sem mindig egyértelmű. Sok esetben a dokumentáció áttekintése a készítőkön kívüli személyek számára szinte lehetetlen feladat [FEI82]. Az utóbbi megállapításon nem is lehet csodálkozni, mivel ahány hibaeset, a dokumentáció annyiféle szervezésére volna szükség. A lehetőségek ebben a tekintetben igen tágak, pl.

- távdiagnózis a hiba felderítéséhez és megelőző karbantartáshoz
- a térbeli vizuális adatbázisok terén elért eredmények fölhasználása
- a dokumentáció probléma és személyfüggő megjelenítése, számítógépes tárolása, manipulálása és a hibakeresés támogatása stb.

A referenciarendszerként kezelt vizsgált rendszerrel egyként kapcsolatot tartó karbantartó robot és karbantartó személy

között folyó intelligens dialóguson keresztül a jövőben a hibakeresést és a javítást minőségileg meg lehet majd változtatni, részben automatikussá tenni.

A gyártórendszerek élete folyamán változik környezetük, ezzel együtt változnak a velük szemben támasztott követelmények is - előbb vagy utóbb módosításukra kerül a sor. A módosíthatóság a rendszereknek éppúgy tervezett tulajdonsága, mint bármely más funkcionális sajátosság. A rendszerépítés a software területén kialakított egy olyan iskolát, amely szerint a rendszereket modulárisan kell tervezni, ahol a modulok környezetüktől elfedik az összes olyan döntést, mely megvalósításukkal ("belsejükkel") kapcsolatos [PAR72],[PAR78]. Ezáltal a modulokon belül véghezvitt változtatások (áttérés nagyobb sebességű egységekre, más technológiára, adatszervezésre) kifelé nem mutatkoznak. Ha ezeket a szempontokat a rendszertervezésnél figyelembe vesszük, akkor biztosan kisebb életciklusköltséggel számolhatunk.

1.7 Az innovációs folyamat tárgyalásának összefoglalása

Ebben a fejezetben az integrált anyag és adatfeldolgozó rendszerek innovációs folyamatáról próbáltunk áttekintő képet adni. A vizsgálat célja az volt, hogy megmutassuk, milyen fő

feladatokból áll az innováció, és ezek végrehajtására milyen eszközökkel rendelkezünk ma. Láttuk az eszközök és a technológia fejlődése, valamint a rendszertechnika közötti összefüggést. A jelenleg levonható következtetések az alábbiak.

- 1.7.1 A rendszerek innovációs folyamatának imént megfestett képe sok üres foltot, ki nem alakult kapcsolatot mutat. Mai tudásunk szerint a folyamat véghezvitele nagyszámú heurisztikus döntést, eseti intelligens megoldást igényel.
- 1.7.2 Törekedni kell arra, hogy a gyártórendszerek szélesebb értelemben vett eszközkészletének fejlesztésére iránymutató hatással bíró gyártórendszerkoncepció fejlődjön ki és terjedjen el. Ugyanekkor ezt a koncepciót folyamatosan fölül kell vizsgálni, az eszközalapok öntörvényű fejlődésének nyomonkövetésével. A fejlesztési irányok kitűzése olyan koncepcióval kell, hogy történjen, mely a részeredményeket rövid távon is hasznosítani képes, de nem akadályozza meg az evolúciós természetű fejlődést.
- 1.7.3 Alkalmas kutatási irányokat és feladatokat kell kitűzni, melyek meglévő eszközök összekapcsolásán,

hiányzó láncszemek pótlásán keresztül javíthatnak az innovációs folyamat minőségén. A folyamat teljes integrálásra törekszünk, de reális lépésekben: csak a kritikus szellemi tőke (emberfő és tudományos eredmény) megszabta körben indítható siker reményében projekt -- ellenkező esetben a szellemi tőkét kell növelni.

- 1.7.4 A folyamat egyik igen lényeges, és a tárgyalás folyamán többször visszatérő eleme a funkcionális tervezés, melynek módszereivel jelen dolgozat foglalkozik. Még ez a szűkebb, viszonylag új szakterület is eléggé széles ahhoz, hogy lehetetlen vállalkozás legyen annak teljes irodalmát áttekinteni és a meglévő módszerek közül az optimálisat kiválasztani. Az egyetlen optimális módszer fogalma amúgy is idegen abban a helyzetben, amikor egészen bizonyos, hogy döntésünket hiányos információ birtokában kell meghoznunk. Ez nem jelentheti azt, hogy nem is választunk, hiszen a módszerek hatékonysága közötti relatív különbség esetleg igen kicsiny a velük elért abszolút nyereséghez viszonyítva. Mindehhez azt sem szabad elfelejteni, hogy a módszertanok alkalmazhatóságának feltételei között igen egyedi megkötések

is találhatóak (pl. a rendelkezésre álló tervezőcsoport, a gazdasági és szellemi mikroklíma stb.).

Összefoglalva: a dogmatikus és diktatórikus módszerek helyett az egyedi helyzethez adaptálható módszereket igyekszünk előnyben részesíteni -- mint ezt a dolgozat 3.1 (konceptió) fejezetében részletesen is kifejtjük.

Ebben a fejezetben a funkcionális tervezés néhány ismert módszerét tekintjük át, és hasonlítjuk össze.

2. A funkcionális tervezés irodalmának áttekintése

Figyelmünket ezután az innovációs folyamat második fő láncszemére, a funkcionális tervezés szakaszára koncentráljuk. Meg kell különböztetnünk a megvalósulási folyamat e szakaszában használt modellezési technikákat és az ezek fölhasználásával született módszertanokat. A modellezési technikák meghatározzák a vizsgálódás körét és fogalmait. A módszertanok egy, vagy több technikát használnak föl valamilyen meghatározott tervezési cél érdekében.

A funkcionális modellek határozzák meg a tervezett rendszer feladatát, elvonatkoztatva a feladatot végrehajtó eszközök egyedi jellemzőitől. A feladatot, mint összetett fogalmat úgy lehet meghatározni, ha visszavezetjük ismertnek feltételezett fogalmakra. Eszerint egy funkcionális modell úgy is tekinthető, mint a leírt rendszer feladatának fogalmi meghatározása. Egy rendszer feladata az általa végrehajtandó transzformációk összessége. Ennek leírásához meg kell határozni a transzformációt mint cselekvést, valamint a transzformáció tárgyát képező

dolgokat. (Nyelvi analógiával élve az állítmányt és a tárgyat.) A modellezés során az idő mint dimenzió nem szerepel - ellentétben pl. a szimulációs modellekkel. Az időben megvalósuló funkcionális kapcsolatok (mint precedencia, vezérlés) azonban a funkcionális modellben természetesen ábrázolhatóak és ábrázolandók is. Az idő dimenziója nem egyszerűen hiányzik, hanem éppen az időbeli általánosítás útján nyert időtől független funkciókat rögzítjük.

Vannak modellek, melyek csak a rendszer logikai állapotai közötti transzformációkat írják le (vezérlési állapotok transzformációja), megengedve a transzformációk konkurrens végrehajtását is. Más modellek a vezérlési állapotokat éppoly dolognak tekintik, mint a többi transzformálandó be- és kimenetet.

Ilyen modellek használatosak pl. a termelési folyamatok (gépipar, vegyipar) analízisére és a számítástechnikai információs rendszerek, software rendszerek tervezésére. A gépipari rendszerek tervezésénél használatos követelményspecifikációs módszereket az alábbi csoportosításban tárgyaljuk:

- a.) grafikus funkcionális modellek előre definiált elemi funkciók nélkül

- b.) grafikus funkcionális modellek előre definiált
elemi funkciókkal

- c.) írott specifikációs nyelvek és adatbázisok
felhasználása.

Az áttekintett irodalom részletesen az alábbi funkcionális
modellezési módszerekkel foglalkozik (I. táblázat).

a.)

HORI cella modell
SADT Struktúrált Analízis és Tervezési Módszer
IDEF az ICAM projekt specifikációs módszere

b.)

Petri háló
AP-háló
GRAFCET háló

c.)

ISDOS PSL/PSA információs rendszerek tervezésére
szolgáló adatbázis
SDLA Rendszerleíró és logikai analizáló
adatbázis rendszer

Az áttekintett funkcionális tervezési modellek
és módszertanok

----- I. Táblázat -----

2.1 Struktúrált analízis és tervezési módszer (SADT)

2.1.1 Az SADT módszer ismertetése és értékelése

Az SADT ([SOF76], [ROS77], [PAL79]) tetszőleges nagy rendszerek struktúrált követelményspecifikációjának előállítására kifejlesztett módszer és módszertan. A segítségével végzett munka eredménye egy grafikus nyelven megfogalmazott, bizonyos minőségi követelményeknek eleget tevő funkcionális rendszerterv.

A modellezés alapfogalmai [ROS77]:

- funkció
- dolog
- mechanizmus

A modellezés alaprelációi:

- bemenet, vezérlési bemenet, kimenet (funkció és dolog között) /ICO/
- rész/egész viszony (funkció és funkció ill. dolog és

dolog között)/ICO/

- mechanizmus/támogatott funkció viszonya (funkció és realizációja között)/M/

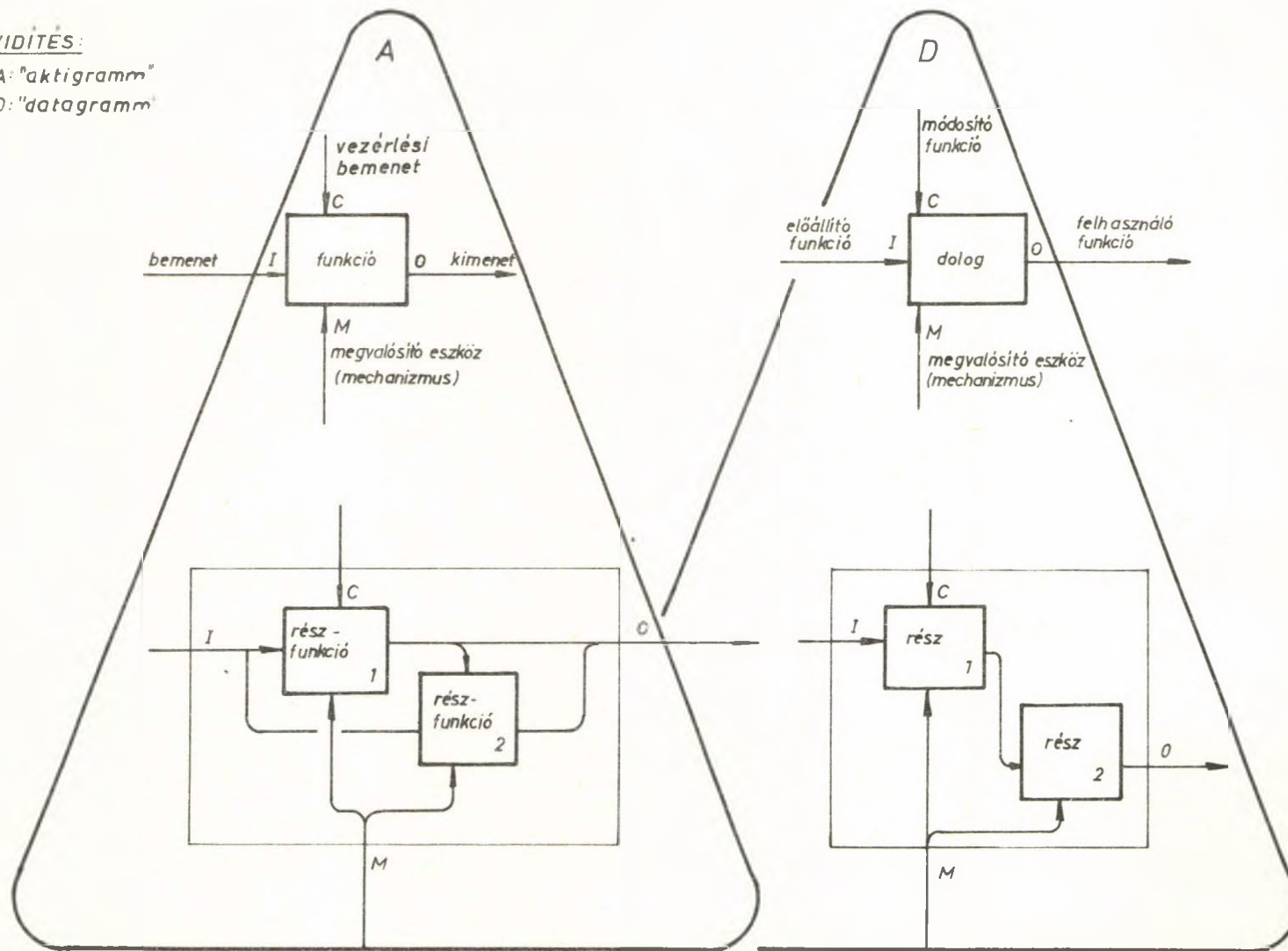
Az SADT módszer szerint egy rendszer leírására több modellt kell készíteni. Egy-egy modell a rész/egész reláció szerint a funkciókat fastruktúrába rendezi, miközben ábrázolja a funkciók és dolgok ICOM relációit. Minden modellel párhuzamosan elkészítendő a dolgok fastruktúrába rendezése is, ezek a modellek úgyszintén leírják az ICOM relációkat (ld. 2.2. ábra). A jelölésrendszer részletes ismertetését itt nem ismételjük meg, mivel az megtalálható [ROS77]-ben.

A modellezés grafikus formában történik, szem előtt tartva az ember általi kezelhetőség és az áttekinthetőség követelményét is. Eszerint egy funkciót (vagy dolgot) nem bontunk 5-7 résznél többre. Az SADT modellekben a funkciók lebontásakor a dolgokat olyan irányított gráf élei ábrázolják, amelynek végpontjai azonosak a funkciók be és kimeneteivel. (Ennek továbbfejlesztését ld. a 4.2.1 pontban.) Ellenkező előjellel ugyanez elmondható a dolgok lebontásáról is. A fastruktúrába rendezés természetes folyamánya, hogy a módszer gyakorlati méretű rendszerek leírását több, egymással összefüggő modell segítségével oldja meg. A SADT módszer segítségével elvileg el

RÖVIDÍTÉS:

A: "aktigramm"

D: "datagramm"



- 2.1 ábra -

lehet érni a lebontásnak arra az alsó szintjére, mely a részfunkciók végrehajtását modellezi. Ez az eset pl. számítógépprogramok tervezése esetén.

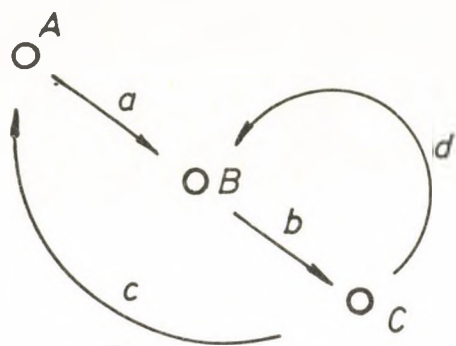
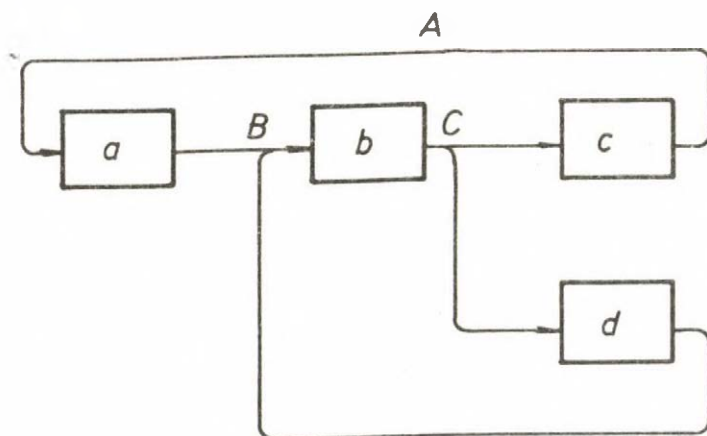
Sokszor a modellkészítés célja nem az, hogy egy rendszert kimerítően leírjunk, hanem olyan összefüggő modellt kell előállítanunk, mely csak az alsószintű funkciókat végrehajtó eszközök összekapcsolásának helyességét hivatott kimutatni. Ez a fontos gyakorlati eset nem igényli az összekapcsolt eszközök belső működésének ábrázolását, csupán az eszközök logikai kapcsolatát. Az SADT módszert fejlesztői elsősorban ilyen -- nagy rendszerek követelményspecifikációjánál jelentkező -- problémák megoldására fejlesztették ki. (U.m. TRAIDEX [TRA76] - katonai oktatási anyagok nyilvántartására, ICAM [AF 78] az USA légihaderő finanszírozta, számítógéppel integrált gyártórendszer projekt lemezszerű tesztek előállítására és szerelésére). Van példa ezenkívül arra is, hogy a SADT módszert kézi szimulációra vagy software tervezésére használták föl - u.m. [SCH78], [DIC78].

Az irodalom a SADT modellezési módszerét mint kézi módszert ismerteti, grafikus nyelve az irodalomban nincs formalizálva. Kézi módszerként használtuk föl a publikációkból kiindulva, részint rekonstruálva, részint továbbfejlesztve a komplex

gépipari rendszerek funkcionális tervezési módszertanának ([BER81c], [BER81d], [BER82c]) kidolgozásánál. A formalizálás hiánya gyakorlati felhasználását közvetlenül nem akadályozza, mert ember-ember közötti kommunikációra így is alkalmas.

Az SADT modellek az idő fogalmát nem használják, amit kritikusai gyakran hoznak föl ellene. Az időbeli általánosítás azonban, mint a funkcionális analízis természetének tárgyalásakor a bevezetőben megvizsgáltuk, az állapottranszformációk szekvenciájának kötöttségeit ábrázolni engedi.

Az állapotgráfok és funkcionális modellek közötti összefüggést a 2.2. ábrán világítjuk meg. Az alapvető különbség az állapotgráf és a funkcionális modell között az, hogy az a,b,c,d funkciók közül az állapotgráf esetében csak az egyik lehet aktív, míg a funkcionális modell esetén egyszerre több is. Ezen túlmenően az A,B,C, állapotleíró vektorok [2.2. ábra] lehetnek parciálisan definiáltak, tehát az állapottranszformáció (funkció) általában nem egy állapotot (mint dolgot), hanem egy állapotosztályt módosít. A funkció maga ugyyszintén nem egy egyszerű állapottranszformáció, hanem állapottranszformációk egy osztálya. A transzformációosztály egy reprezentánsa (mely az időben keletkezik) az állapotosztály egy reprezentánsát (egy adott állapotot) transzformálja. A funkcionális modelle-

állapotgráffunkcionális
modell

zés állapot- és transzformációosztályok bevezetésén keresztül vonatkoztat el az időtől. A SADT módszertana nemcsak kész modelleket vizsgál, hanem a modellek előállításának folyamatára is tesz ajánlásokat, midáltal nemcsak a kész modellek, hanem a közbelső termék minősítési és kezelési módszereit is taglalja.

2.1.2 A SADT módszer kritikája

A funkcionális modellek grafikus ábrázolási nyelve általános, könnyen érthető. Segítségével több szakterület képviselőinek egyértelmű kommunikációja oldható meg. Ezen keresztül alkalmazása a gyártórendszerek tervezésében egy igen fontos nehézséget old meg, a csatlakozó felületek definiálásának kérdését.

A módszer általános, mivel nem köti meg a tervezés során alkalmazható fogalomkészletet, csupán a funkcionális modellezés kategóriáit és azok relációit. Ez a tulajdonsága más, hasonló célú módszerekkel szemben pozitívan értékelendő.

A modellezési módszer kézi úton is hatásosan alkalmazható, így viszonylag kis befektetéssel adaptálhatja egy tervezőkollektíva. Amiképpen a kézi alkalmazhatóság előny, ugyanúgy hátrány is, mivel a tervezett rendszerek méretének növekedésével egyre

szükségesebbé váló automatikus ellenőrzési lehetőség nincs /nem volt/ kidolgozva.

Az automatizált modellépítés és ellenőrzés hiánya miatt nem merült föl a modellezési módszer formalizálásának kényszere, pedig a felhasználás határainak ismerete, továbbá az érdemi ellenőrzés nélkül nem oldható meg.

A módszer tisztán funkcionális modelleket állít elő. Az idődimenziótól történt elvonatkoztatást lehet mind pozitív, mind negatív előjellel értékelni, attól függően, hogy milyen fejlesztési stratégiát és célt kívánunk az eljárással támogatni. Ha a modellezés célja a követelményeket legjobban kielégítő funkciók és kapcsolataik tervezése, akkor kifejezetten előnyös a funkcionális tervezés időszakában csak a kötelező sorrendiségeket és kényszereket előírni, mert a műszaki tervezés során annál nagyobb lehetőség van a megvalósítási alternatívák közötti választásra. A megvalósítási lehetőségek korlátozottsága az, ami arra kényszeríthet, hogy a tisztán funkcionális tervezést az időbeli vagy más mennyiségi tervezéssel együttesen végezzük, mivel ilyen esetben az eszközkészlet erősen visszahat a tervezhető funkciókra, közöttük a tervezett szekvenciális kötöttségekre is. A módszer továbbfejlesztése gyakorlati tapasztalatok alapján részben választ adott arra a

kérdésre (4.1.4.1. fejezet), hogy miképpen lehet a kötöttségeket minimalizáló eszközkészletet létrehozni.

A SADT módszer hiányossága, hogy nincs kidolgozva a segítségével készített funkcionális modellek és realizációjuk formális kapcsolata. Mivel éppen a realizációtól elvonatkoztató tervezés céljából használjuk a módszert, nem a realizáció algoritmizálását hiányoljuk, hanem a modellek és a realizáció fogalmi kapcsolatának tisztázását. A tervezés viszonylagos realizációfüggetlensége megengedi, hogy eredményének többfajta megvalósítását állítsuk elő, ennek ellenére szakterületenként hasznos volna egy vagy több megvalósítási stratégiát (technológiát) kidolgozni, melyeket azonos funkcionális tervből kiindulva mennyiségi szempontok szerint lehetne esetenként mérlegre tenni.

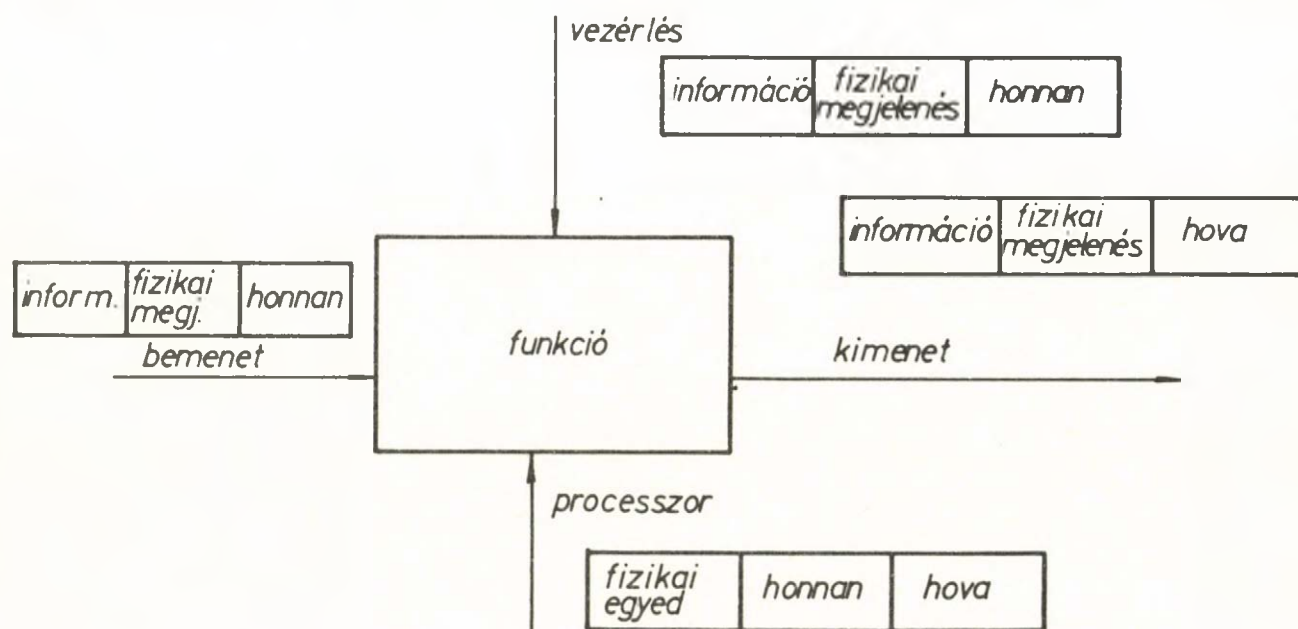
Az SADT modellek eredeti publikációiban [SOF76], [ROS77] a mechanizmus fogalma körül fogalmi zavar van. A módszer megalkotói, látván a köznapi értelemben vett mechanizmusok ábrázolásának szükségességét, (Hori "processzor"-a [HOR72]), szembe kerültek a mechanizmusok kétarcúságával: egy mechanizmus (eszköz) rendelkezik dologi és funkcionális természettel egyaránt. A formalizálás hiánya miatt kénytelenek voltak a mechanizmust mint harmadik kategóriát bevezetni. A gyakorlatban ez nagyobb

problémát nem vet föl, csupán akkor, ha a realizációval fennálló fogalmi kapcsolatot próbáljuk megfogalmazni. A kérdést öndllő eredmények alapján a dolgozat 4.2.1.4.2. fejezete tárgyalja.

Itt kell megemlíteni, hogy a SADT-ről igen kevés nyilvános publikáció született, sok közvetett információ és gyakorlati kísérletek alapján kellett rekonstruálni illetve továbbfejleszteni, a know-how vásárlására tett kísérlet pedig embargó miatt meghiúsult.

2.1.3 A SADT modellezési módszer történeti fejlődése

A SADT elődje Hori cella modellje [HOR72], mely az alábbi fogalmakat definiálta (ld. 2.3. ábra). A cella modell fogalmainak szükségességét Hori a termodinamika II. főtételenek



cella modell

alapján magyarázza. Ennek lényege, hogy az emberi cselekvés vagy ember által létrehozott folyamat a valószínűbből a kevésbé valószínűbe viszi át a rendszert. A funkció a bemenetekből (a lehetséges állapotokból) előállít egy kevésbé valószínű állapotot, a kimenetet. Ahhoz, hogy az a II. főtétellel ne legyen ellentmondásban, szükség van információra (vezérlés) és energiára, mely energiát valamely fizikailag megjelenő egyed kell, hogy közvetítse (processzor).

Mind a be- és kimenet, mind a vezérlés rendelkezhet informatív és anyagi természetű attribútumokkal, míg a processzor szükségszerűen fizikai egyed. Ilyen értelemben a vezérlő információ a kimenet szimbolikus megfelelője. A funkció egy fizikailag lejátszódó folyamatot jelöl, mely energiát közvetít és információt termel. A folyamat térben és időben játszódik le, és negatív entrópiát hoz létre. A vezérlő információ az energiaközvetítő processzort irányítja. A bemenet egy egyedére valójában az így közvetített energia hat és így hozza létre a kimenetet. D.T.Ross az SADT kidolgozásánál az itt megjelenő fogalmakat általánosította az alábbi módon:

- Az információ helyére lép a dolog fogalma (az eredeti SADT publikációkban data, de tartalmi fordítása nem adat, hanem dolog). A funkciókhoz hasonlóan

a SADT ezt a fogalmat is struktúrálja, és információt vagy anyagot, valamint azokból képzett osztályokat egyaránt ábrázolhatunk segítségével.

- A fizikai megjelenés ábrázolásának igényéből jött létre a mechanizmus fogalma, tehát nemcsak a funkciók, hanem a dolgok is rendelkeznek mechanizmussal.
- A honnan-hova relációk a diagramkészítési módszer szerint grafikus formában jelennek meg.
- A dolgok struktúrálása szükségessé tette a nyílhálózat elágaztatásának értelmezését, továbbá a dolgok explicit struktúrájának bemutatására a datagrammok bevezetését.
- A struktúrált lebontás korlátait csak a modellfogalom bevezetésével lehetett átlépni, így t.i. a fastruktúra nem megkötés többé. Megjelent a rendszerleírás és a modell fogalmának elkülönítése.
- Az SADT modell letisztult abban az értelemben, hogy a mennyiségi és realizációs fogalmaktól megszabadí-

totta a funkcionális modelleket [ROS77].

- A SADT módszert adott a kezelhető méretű és emberileg áttekinthető modellek készítésére [SOF76].
- Mindezeket az előnyöket gyakorlati példákon igazolta [AF 78], [TRA76].

Több kísérlet történt a SADT fogalomkészletének kibővítésére (részint azonos [MUL78], részint különböző [GAN79],[MAR78] grafikus notációval). Általában egyes szakterületek indíttatására specialitásokat vittek a funkcionális analízisbe -- részint a modellek tisztán funkcionális mivoltát megszüntetve. Gane-Sarson módszere pl. a software tervezés területén alkalmazza a SADT aktigramkészítési módszerét, az adatstruktúrákat azonban implementációs aspektusokkal keveri (különböző volatilitású adattárak más fogalmi besorolást kapnak). Ez tisztán software rendszerek tervezése szempontjából megengedhető, sőt felette hasznos lépés -- előrevetítve annak a gondolatát, hogy a rendszertervezői rendszerek nem diktatórikus, hanem permisszív tulajdonságokkal rendelkezzenek. A követelmény több módszer ötvözése, összekapcsolása az egyes feladatoknak legmegfe-

lelőbb módon.

A CORE módszertan [MUL78] a SADT-nek szinte csak az ábrázolási formáját tartotta meg, megjelenítve az idő fogalmát. Így a CORE két módszernek, a funkcionális tervezésnek és a blokkkészítésnek a kereszteződése. Szintén software, mégpedig elsősorban valósidejű programok tervezésének igénye motiválta a szerzőt a két módszer összeolvasztására.

A lemezszerű testeket (pl. repülőgépelemeket) gyártó gyártórendszer létrehozására irányuló nagyszabású ICAM program [AF 78] (Integrated Computer Aided Manufacturing) szintén az SADT módszerre alapozva dolgoztatta ki az ICAM gyártórendszer architektúráját. A teljes ICAM projektben IDEF0 néven vezették be a SADT modellezési módszerét az ezt kiegészítő számítógépes rendszerrel együtt. Az alábbiakban az ICAM projekt mellé rendelt és az NAS (1) keretében működő szakmai felügyelő bizottság (COCAM) jelentéséből idézünk -lévén, hogy részletesebb elemzésre alkalmas anyag nem áll rendelkezésünkre [COC81].

A jelentésből kitűnik, hogy jól mértük föl 1976/77-ben azt az igényt, miszerint a SADT módszerhez nagy projektek

(1) USA, National Academy of Sciences

modelljeinek kezelésére számítógépes támogatás szükséges. Ugyanekkor a választásnál korszerűbb és gazdaságosabb megoldás felé indultunk el -- bár választásunkban a lehetőségek determináló ereje sem volt kevés. A rendszertervezést támogató számítógépi rendszer általunk kidolgozott struktúrája igen kedvezően ítéltető meg a jelentés alábbi részleteinek ismeretében.

"Az ICAM program igen értékes képet kapott a gyártás architektúrájáról. Az architektúra közös nyelve (az ICAM definíciós nyelv, vagy IDEF) nagyban megnöveli az ipari, kormány és kutató körök közötti kommunikáció és együttműködés hatékonyságát." [COC81 10.3 bek.]

"Az ICAM program eredményeinek felhasználása:

Néhány ICAM termék már használatban van, és hasznot hajt az ipárban. Például az ICAM program fejlesztette ki az IDEF módszert a funkciók definiálására. Ezt az ábrázolási módszert ma szabványként használják az ipar egyes ágaiban és használata folyamatosan terjed." [COC81 7/2 bek.]

"Integrációs stratégiák:

Az ICAM program célja, hogy a gyártási műveletekhez és irányításukhoz egymással szisztematikus kapcsolatban lévő rendszermodulokat fejlesszen ki az igazán nagy hasznót azonban csak akkor fogja meghozni, amikor az önállóan kifejlesztett CAM elemeket egy nagy integrált rendszerbe fogják össze.

IDEF:

Az ICAM definíciós módszere három egymást támogató eszközből áll, melyek fejlesztési állapota jelenleg eltérő [megj. az idézet 1981 júniusi keltezésű]. Ezek: Az IDEF0 a funkciómodellezésre, az IDEF1 az információmodellezésre és az IDEF2 a dinamikai modellezésre.

Az IDEF0 grafikus módszer a gyártási funkciók analízisére készített diagrammok és modellek létrehozására,

szerkesztésére, megjelenítésére, kirajzolásra és verifikálásra szolgál. Az információs modell, az IDEF1 a közelmúltban került használatba: ez az eszköz dokumentálja az információfolyamot és az integrált rendszerek softwarejének fejlesztéséhez fog segítséget nyújtani. Az IDEF2 dinamikai modell az IDEF eszközök legújabbika, jelenleg fejlesztés alatt áll, és felhasználási módja még tisztázatlan....

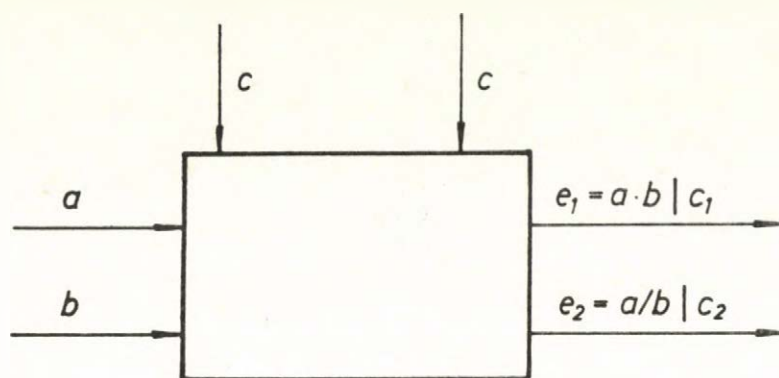
Az IDEF szolgáltatások sokkal értékesebbé válhatnak, ha automatizáltsági fokukat megnöveljük és felhasználhatóságukat megkönnyítjük. Fontos, hogy ezeket az eszközöket olyan módon fejlesszük ki, hogy a gépészmérnökök minden további nélkül alkalmazhassák a követelmények definiálására és dokumentációra." [COC81 14/5 bek.]

"Az IDEF eszközök számítógépköltsége:

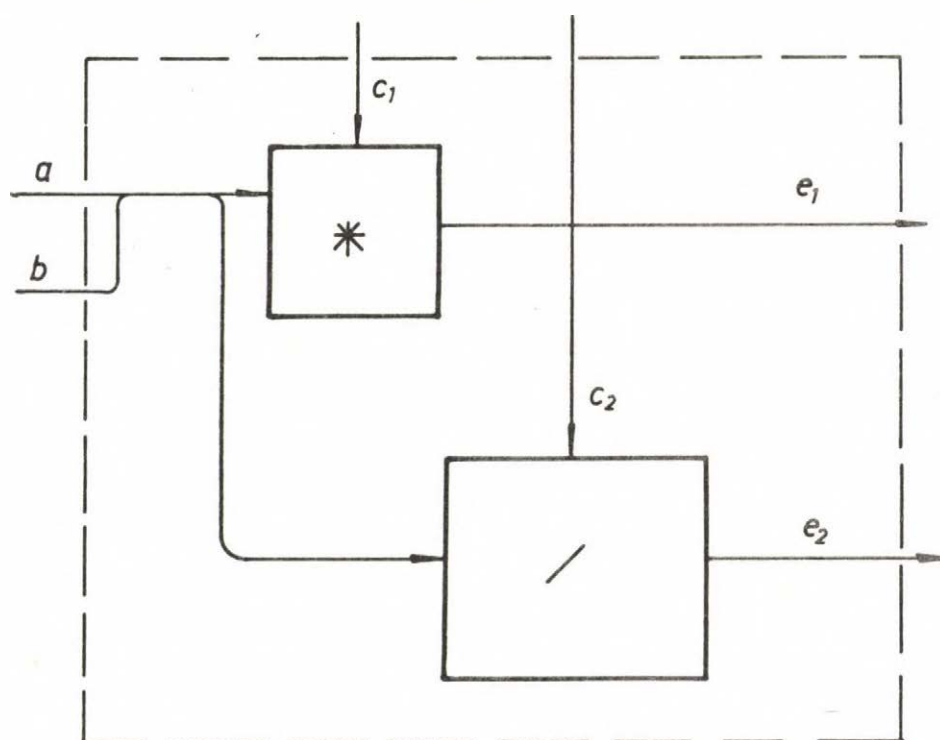
Az IDSS [Interactive Decision Support System -- az ICAM döntéstámogató rendszere, benne az IDEF0 és IDEF1 eszközökkel] az adatgyűjtési és modellezési problémát igyekszik megoldani azáltal, hogy hozzáférést ad a gyártásleíró adatbázishoz és grafikus csatlakozó felületéhez.....

Az IDSS számítógépes nemzeti hálózaton (CYBERNET) keresztüli használatnak jelenlegi költségei még a tervezett új képességek nélkül is nagyok.... A programirodának és szerződő feleinek alternatívát kell keresniük az információelosztás módszerére, hogy a felhasználók saját számítógéprendszerüket használhassák...." [COC81 16/3 bek.]

Kísérletek történtek tisztán funkcionális modellek és softwarekészítési módszerek kapcsolatának létrehozására, különféle annotációk bevezetésével [DIC78] (ld. 2.4/a ábra). Igen egyszerűen belátható, hogy ilyen jellegű notációk a lényegen nem változtatnak, a bemeneti és kimeneti kapcsolatok ilyen föltüntetése csupán írásmunkát takarít meg (egy szinttel kevesebbet kell diagrammon ábrázolni -- ld. a 2.4/b helyettesítő ábrát).



fünkcionális diagram annotálása (a)



és amit helyettesít (b)

Egy különbség mégis található: nevezetesen az annotáció nyelve (nevezetesen az adott programozási nyelv utasításai) a funkcionális tervbe realizációs döntéseket visznek. Ha azonban a rendszermodell megvalósításában szigorúan a funkcionális felépítést tükröző fizikai struktúrát használunk, akkor a módszer jogosult. (Ugyanis, ha a funkcionális és realizációs struktúra nem azonos, nincs annotálásra /1-1 megfeleltetésre/ lehetőség.)

A SADT egyik fontos tulajdonsága, mely általános alkalmazhatóságának legfőbb eleme, hogy nem használ előre kötött primitív elemkészletet. A következőekben olyan közismert módszereket mutatunk be, melyeknél adottak a primitív funkciók, vagy tulajdonságaik legalábbis részben kötöttek -- függetlenül attól, hogy milyen típusú rendszert modellezünk segítségükkel.

2.2 A Petri háló alapú modellezési technikák

2.2.1 Petri háló (P-háló)

A Petri háló (P-háló) konkurens rendszerek állapotátmeneteinek modellezésére mind szimuláció mind matematikai analízis révén alkalmas, t.i. minden alapelemének ismert a viselkedése. A P-háló vizsgálatának során definiálni lehet néhány alapvető tulajdonságot, mely gyakorlati problémák megoldásánál jól használható [RAM80].

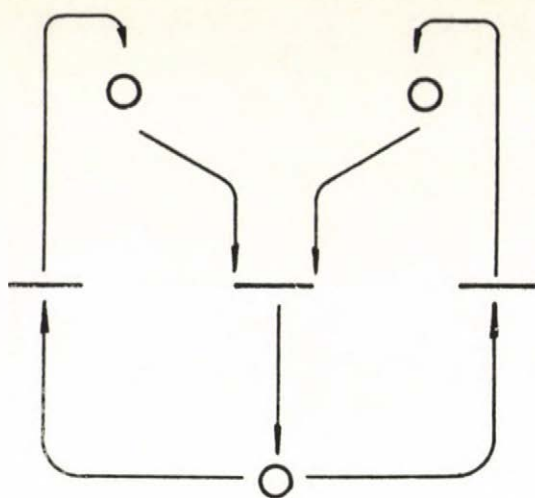
A Petri háló grafikus képe olyan irányított gráf, melynek élei a kimenete és bemenete relációt ábrázolják, szögpontjai

pedig a helyek és átmenetek. Minden él pontosan egy helyet köt össze pontosan egy átmenettel. A helyeket körökkel, az átmeneteket vonással szokás ábrázolni (2.5. ábra).

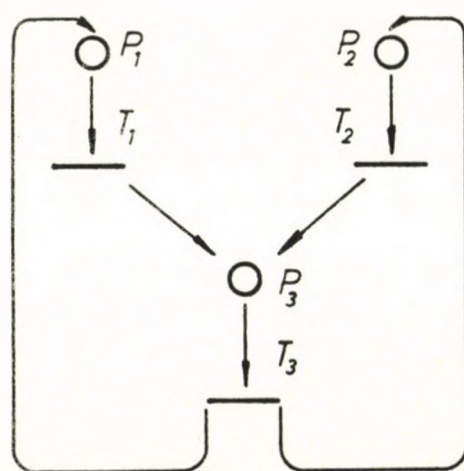
A Petri hálók olyan funkcionális modellek, melyek konkurrens rendszerek leírására szolgálnak [PET78]. Ismertetésüket nem a szokásos matematikai módszerekkel kezdjük, hanem a vizsgálat tárgyát képező funkcionális analízis szemszögéből.

A Petri hálók ismert viselkedésű alapépítőkövekből épülnek föl, így a segítségükkel leírt rendszerek működése is minden pillanatban ismert. A két fogalom, a funkció és a dolog itt is megtalálható, és pedig a funkció fogalmának megfelel az átmenet, a dolognak pedig a hely fogalma. Közöttük a "bemenete" és a "kimenete" relációk állnak fenn. Nincs funkciók és funkciók illetve dolgok és dolgok közötti reláció -- a Petri hálók nem struktúráltak. Az alapelemek jeleken keresztül fejtik ki viselkedésüket. A hely alkalmas jelek tárolására, az átmenet pedig jelek továbbítására. Az átmenet egy-egy jelet továbbít minden kimenetére, ha minden bemenete legalább egy jelet tárol.

Az apróbetű röviden a P-hálók matematikai definícióját mutatja be, mely a P-hálók ihlette módszerek és a P-hálók összehason-



- 2.5 ábra -



- 2.6 ábra -

lításához lesz hasznos jelen dolgozatban.

legyen $P (P_1, \dots, P_i, \dots, P_n)$ a helyek halmaza
 $T (T_1, \dots, T_j, \dots, T_m)$ az átmenetek halmaza
 $I = P \times T$ a bemeneti ágak incidenciamátrixa
 $O = T \times P$ a kimeneti " " "
 Itt $M (M_0, \dots, M_n)$ a P-háló egy jelkiosztása, ahol
 M_i az i -edik helyre eső jelek száma.
 $TR_k = (TR_{1k}, \dots, TR_{jk}, \dots, TR_{mk})$ a k -adik pillanatban
 aktivizált átmeneteket írja le: $= 1$ ha aktív
 $= 0$ ha nem
 $M_k (M_{1k}, \dots, M_{ik}, \dots, M_{nk})$ a P-háló k -adik állapota
 ahol M_{ik} egész ≥ 0
 ekkor $TR_{jk} = 1$ ha $I_{ji} \neq 0$ és minden i -re ($I_{ji} = 0$ vagy
 $I_{ji} \cdot M_i \neq 0$)
 $= 0$ egyébként
 $M_k' = M_k - I \cdot TR$
 $M_{(k+1)} = M_k' + TR \cdot O$
 Konfliktus lép föl, ha M_k' valamelyik eleme negatívvá
 válik, ezesetben az algoritmus kivezet az M állapotteré-
 ből ($M_{ik} < 0$).

Egy adott jelkiosztást a P-háló egy állapotának nevezünk.
 Bármely kezdeti jelkiosztás definiálja jelkiosztások egy so-
 rát, mely halmaz a P-hálónak az adott kezdeti állapothoz
 tartozó elérhetőségi halmaza. Az elérhetőségi halmaz ismerete
 gyakorlati jelentőséggel bír. Úgyszintén érdekes tulajdonság a
 P-háló élő mivolta. Egy háló él, ha elérhetőségi halmazának
 bármely jelkiosztásból elérhető a halmaz valamely másik
 jelkiosztása olyan úton, mely egy előre definiált (tetszőle-
 ges) átmeneten keresztül vezet (bármely funkció aktiválható).
 Ilyen problémák vizsgálatára van szükség például több hozzáfér-
 őt kiszolgáló rendszerek erőforráselosztó algoritmusainak

elemzése során. További alapvető fogalmak a háló tulajdonságával kapcsolatban: két átmenet konkurrens, ha nincs közös bemenetük és konfliktusban van, ha van közös bemenetük.

A P-hálók alapvető problémája az elérhetőség. Bár sok gyakorlati probléma vezethető vissza ennek megoldására, sajnos nagy számításigényű művelet (exponenciális) [LIP76].

A P-hálók gyakorlati használatára többféle kiterjesztéssel éltek. Alapvető kiterjesztés az inhibitor bemenetek bevezetése (az állapotátmenet csak akkor mehet végbe, ha a bemeneten nincs jel), valamint az átmenetek közötti prioritások és az időlimittel rendelkező átmenetek definiálása.

A P-hálók analízise jelentősen megkönnyíthető, ha struktúrális korlátozásokat írunk elő. Ezek betartásával eleve csak a kívánt tulajdonságú modellek készülhetnek. Az így definiált P-háló osztályok:

- véges automaták (minden átmenetnek csak egy be- illetve kimenete van).
- jelölt hálók (minden hely csak egy átmenet be- illetve kimenete). Jelentőségük, hogy eleve

konfliktusmentesek.

- szabad döntési hálók (a jelölt hálóknál valamivel általánosabbak, csak a közös, konfliktust okozó bemeneteket zárják ki) - ld. 2.6. ábra.

A P-hálók egyéb kiterjesztéseinek kiterjedt irodalma van pl. [NOE78], [CZU82]; ld. még [PET78] irodalomjegyzéke.

2.2.2 AP hálók (Abstract Process Nets [MEK80])

2.2.2.1 Az AP-hálók rövid ismertetése

Az AP-hálók struktúrált, konfliktusmentes hálók felépítésére születtek. Lényegük, hogy a struktúrált programozás alapelemeiből kiindulva (szekvencia, döntés, iteráció) adnak általános leírási módszert axiomatikus alapokon. A struktúrálás következményeként a P-hálók "hely" fogalma helyett (mely az adott rendszerállapot-komponensnek felelt meg) be- és kimeneti állapothalmazokat vezetünk be. A SADT-hez hasonlóan az AP-hálók ezen természetük miatt csak a lebontás elkészültével hajthatóak végre. Egy transzformáció be- és kimeneti állapothalmazát predikátumokkal írhatjuk le (precondition, postcondition).

Az AP-hálók formális modellje:

Elemek:

P : absztrakt folyamat

$I(P), O(P)$: be- és kimeneti halmaz

$A(P), Z(P)$: be- és kimeneti állapothalmaz

Elemek kompozíciója:

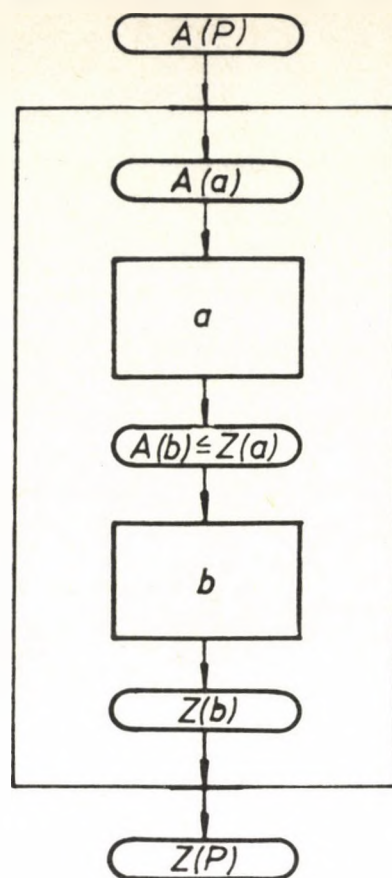
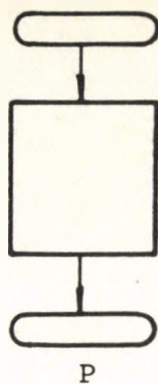
- soros - ha a, b absztrakt folyamatok, akkor $P = ab$
és $A(P) = A(a)$, $Z(P) = Z(b)$, $Z(a) \supseteq A(b)$
- szelektív dekompozíció - $P = a + b$
 $A(P) = A(a) \cup A(b)$, $A(a) \cap A(b) = \emptyset$
- iteratív dekompozíció - $P = a^*b$
- konkurrens dekompozíció - $P = a \parallel b$
- rekurzió - $P = aP + b$

A különböző dekompozíciók grafikus jelölését a (2.7. ábra) szemlélteti.

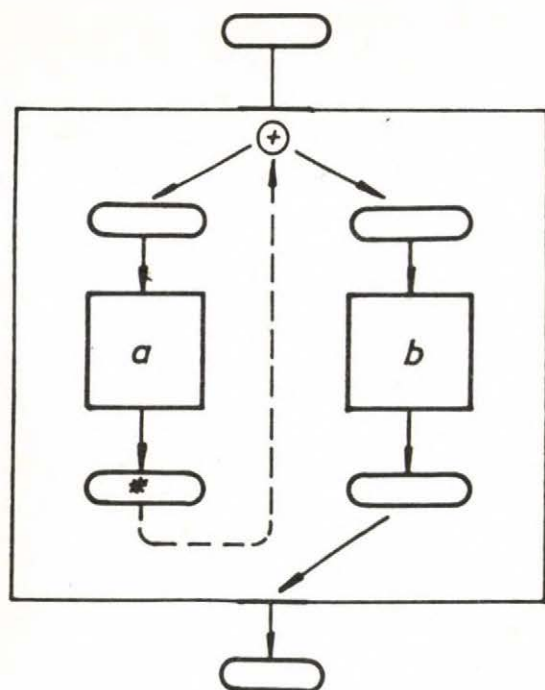
2.2.2.2 Az AP-hálók és Petri hálók viszonya

Az AP-hálók specialitásai:

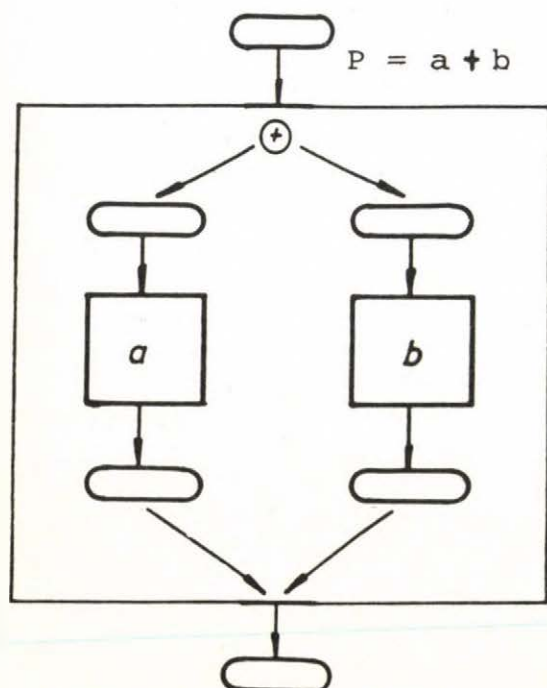
- Az állapotdefiníció átmenettípus bevezetése, mely az állapotátmenetet külön definiálandó szabályhoz köti. Ezáltal kiküszöböli a P-hálók konfliktushelyzeteit.
- A modellezett rendszer állapota és a háló állapota másképpen függ össze, mint a P-hálók esetén, mivel itt az egyes helyek predikátumokban definiált állapotvektoroknak és nem a teljes rendszerállapotvektor egyes elemeinek felelnek meg.



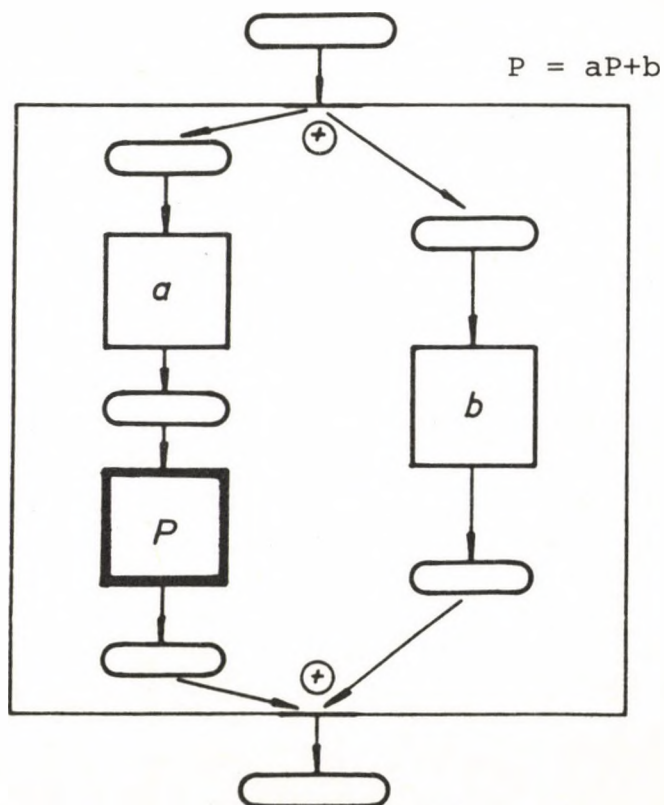
$$P = ab$$



$$P = a * b$$



$$P = a + b$$



$$P = aP + b$$

A dekompozíció során ér el a modellezés arra a szintre, ahol a rendszerállapotvektor komponensei és az AP-háló állapotváltozóinak egybeesnek. A lebontás legalsó szintjén tehát az AP-háló struktúrált Petri háló (2.8. ábra).

2.2.3 GRAFCET-háló [ADE79]

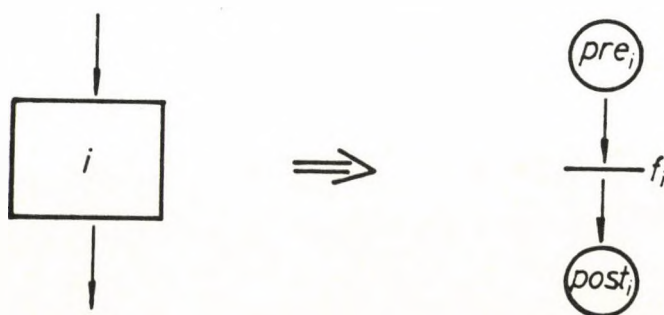
A GRAFCET-háló szekvenciális automaták funkcionális modellezésére születtek, kidolgozói az ipari (gyakorlati) alkalmazhatóság kritériumát tartották szem előtt. Eredeti formájában nem struktúrált modellezési technika. A GRAFCET modell az AP-hálókhoz hasonlóan a P-háló és a funkcionális modellek fogalomrendszere között teremt kapcsolatot. A GRAFCET háló a helyekhez funkciókat, az átmenetekhez feltételeket rendelnek. Formális modelljük fölírható a következőképpen: (az elemzés kedvéért -- ugyanúgy mint az AP-háló ismertetésénél tettük -- elvégeztük a formalizálást).



- 2.8 ábra -



- 2.9 ábra -



- 2.10 ábra -

Helyek: $P (p_1, \dots, p_i, \dots, p_n)$
 Átmenetek: $T (T_1, \dots, T_j, \dots, T_M)$
 $I = P \times T$ a bemeneti ágak incidenciamátrixa
 $O = T \times P$ a kimeneti ágak incidenciamátrixa
 Funkciók: $F (F_1, \dots, F_i, \dots, F_n)$
 Szakaszok: $E (E_1, \dots, E_i, \dots, E_n)$
 ahol $E_i = \langle p_i, F_i \rangle$ a helyek és
 funkciók összerendelése
 Feltételek: $C (C_1, \dots, C_j, \dots, C_m)$
 Állapottranszformációs predikátumok:
 $R (R_1, \dots, R_j, \dots, R_m)$
 ahol $R_j = \langle T_j, C_j \rangle$ az átmenetek és felté-
 telek összerendelése

Ha TR_k ($TR_{1k}, \dots, TR_{jk}, \dots, TR_{nk}$) az állapotátmenetek
a k -edik pillanatban
és M_k ($M_{k1}, \dots, M_{kj}, \dots, M_{kn}$) a k -edik állapot
és M_0 ($M_{01}, \dots, M_{0j}, \dots, M_{0n}$) a kezdeti állapot
akkor
 $TR_{jk} = 1$ ha a./ ($I_j < 0$) és
b./ minden i -re ($I_{ji} = 0$ vagy
 $I_{ji} * M_i \neq 0$) és
c./ C_j feltétel teljesül
= 0 egyébként
Az állapotátmenetek :
 $M_k' = M_k - I \times TR$
 $M_{(k+1)} = M_k' + TR \times O$
Konfliktushelyzet itt is lehetséges, ha létezik
 $M_k' < 0$.
Egy F_i funkció aktív a k -edik időpillanatban, ha
 $M_{ki} > 0$.

Amint látható, a GRAFCET-háló és a funkcionális modellek illetve a Petri-háló összerendelése nem lehetséges egy lépésben, mert mind az átmenetek, mind a helyek funkciókat ábrázolnak. (Az átmenetek döntési, a helyek egyéb funkciókat.) Például a következő illusztráció (2.9. ábra) egy GRAFCET-háló részlete [ADE79].

A GRAFCET-hálók és a Petri-hálók egy-egy megfeleltetését úgy találjuk meg, ha meggondoljuk: a funkcióknak mindig kell, hogy

legyen előzetes végrehajtási feltétele (feltételállapot, precondition) és végállapota (eredmény-állapot, postcondition).
 Eljünk az alábbi összerendeléssel (2.10. ábra). Ekkor a megfelelő Petri-háló a következő helyettesítéssel generálható egy GRAFCET-hálóból (a Petri-háló jellemzőit "++"-al jelölve:

```

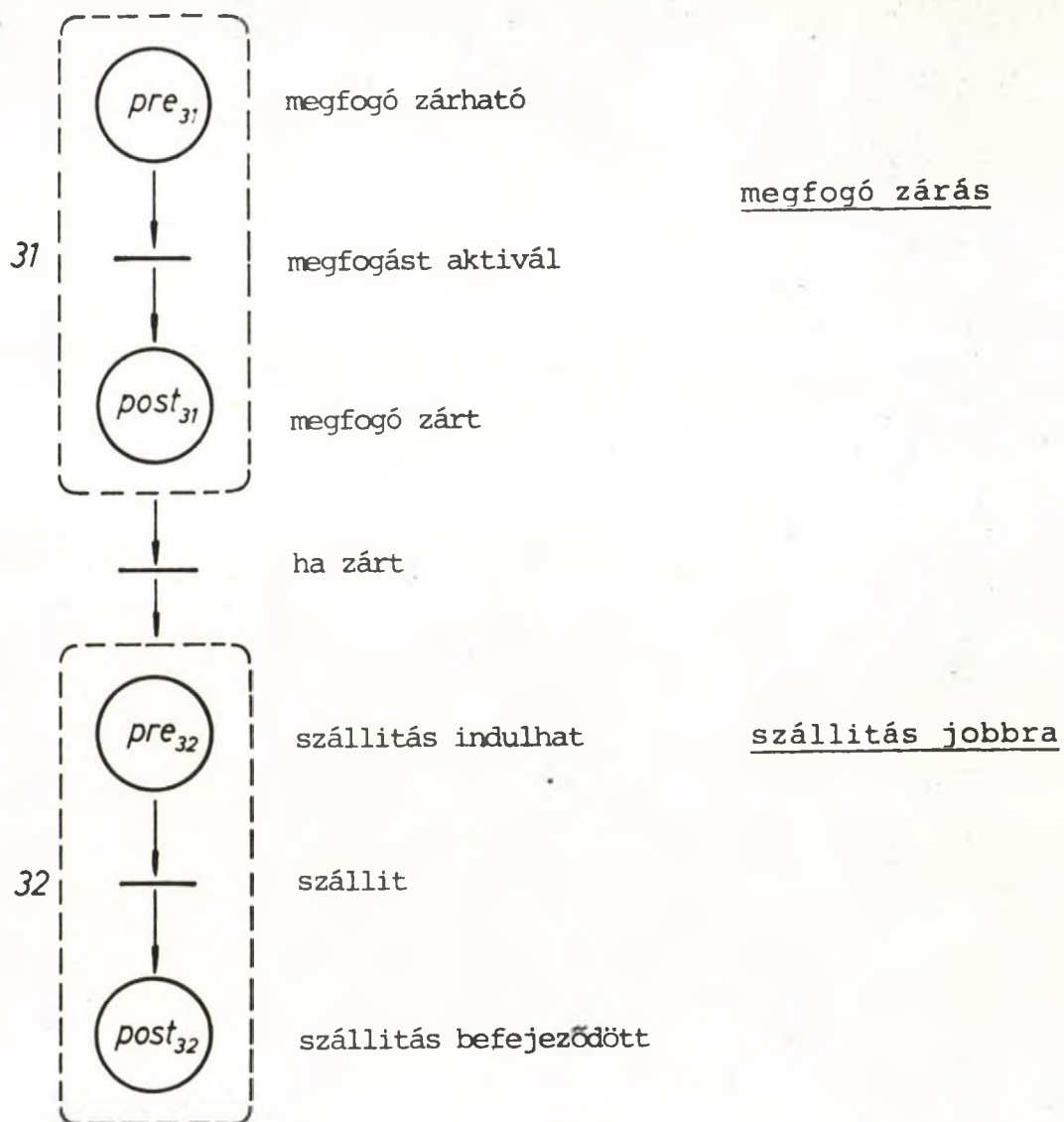
++n = 2 x n
++m = n + m
++P2i = PREi
++P2i+1 = POSTi
++Tj = Tj ha j ≤ n
      = Fi ha j > n (ahol i = j-n)
++I = I elemei + PRE-F ágak
++O = O elemei + F-POST ágak

```

Az állapotátmenetek definíciója a Petri-hálókkal konform, a funkciók aktiválásának modellezése azonban lényegesen precízebbé válik. Ha a PREi jelet kap, akkor a funkciót modellező átmenet aktiválódik, majd azt a funkció lezajlása után POSTi-nek továbbítja. A 2.9. ábra példája a 2.11. ábra szerint módosul.

Az elő- és utófeltételek bonyolultabb predikátumokat is tartalmazhatnak, melyek esetleg egy összetett állapotot határoznak meg. Mindennek tudatában a GRAFCET jelölésrendszere elmentmondásmentesen kezelhető funkcionális modellek reprezentációjára.

A GRAFCET-háló kiterjesztése struktúráldás irányába [MOA81] is



- 2.11 ábra -

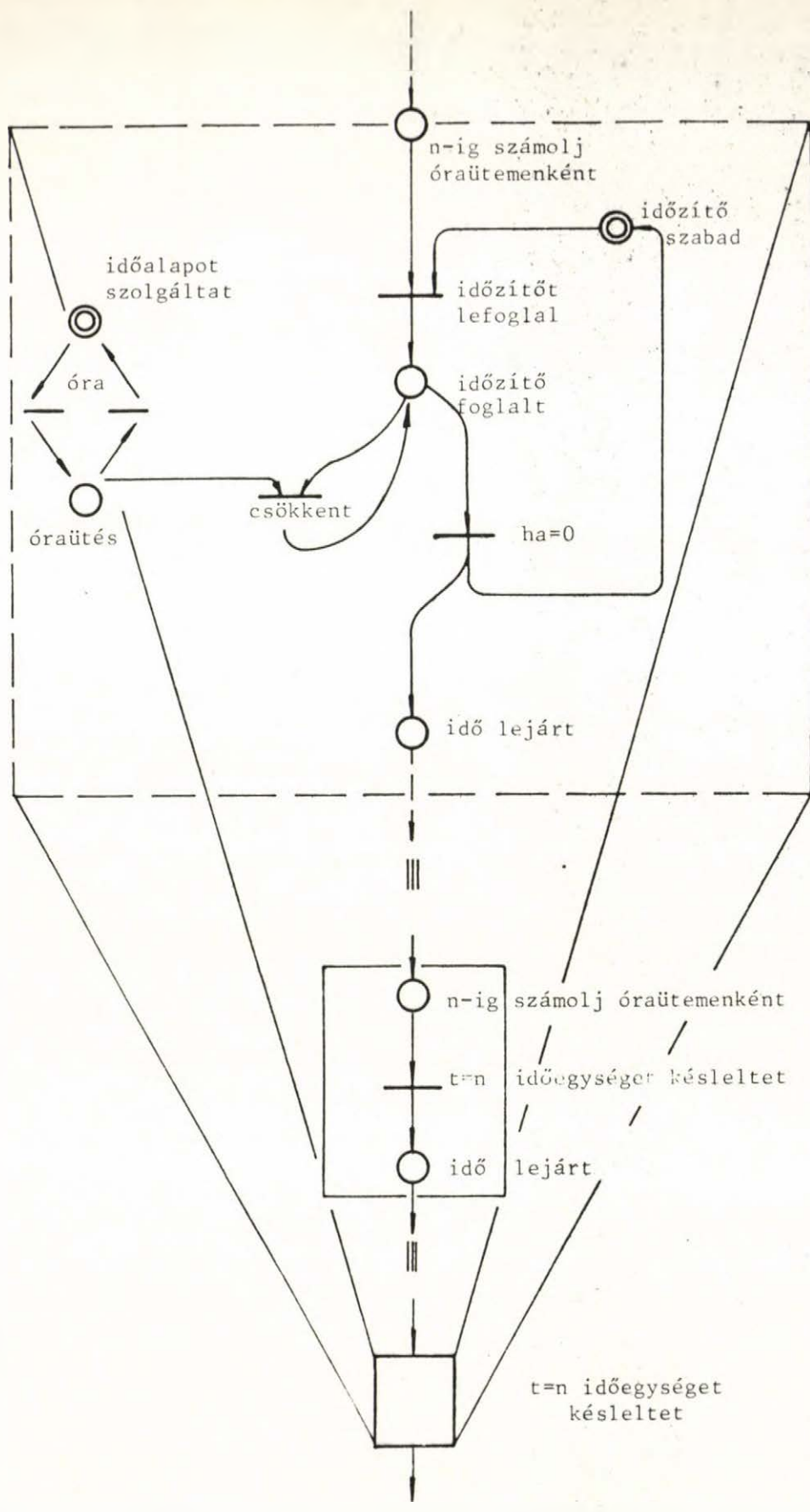
ebben az irányban haladt, mégpedig a makroszakasz fogalmának definiálásán keresztül. Egy makroszakasz egy be- és kimeneti szakasszal rendelkező GRAFCET-háló, mely az eredeti háló egy szakaszának helyébe léphet. Az alábbi példa (2.12. ábra) a GRAFCET-szakasz, a makroszakasz és a Petri-háló összefüggését ábrázolja egy példán keresztül. A modell egy olyan időzítő automatát ír le, mely időzítési parancsot fogad és az idő lejártával jelez. Több parancs sorbadll. (Nem túlságosan gyakorlatias eset, de ez a példa szempontjából közömbös.)

Az előzőekben megmutattuk, hogy az AP-hálók és a GRAFCET-hálók modellezési ereje hasonló. Különbség közöttük, hogy míg az AP-hálók csak a struktúrált programozás alapelemeire épülnek, a GRAFCET-hálók tetszőleges funkcionális struktúrát engednek meg. A valóságban a kezelhetőség követelménye miatt a GRAFCET-hálókat is struktúrált kivitelben helyes készíteni.

2.2.4 A Petri és SADT alapú modellek összehasonlítása

Az előző fejezetben Petri hálóval modellezett funkcióegyüttest SADT-ben modellezve a következő ábrákon lehet nyomonkövetni (2.13., 1.14., 2.15. ábra).

A Petri és a SADT alapú modellek összehasonlításából kitűnik,

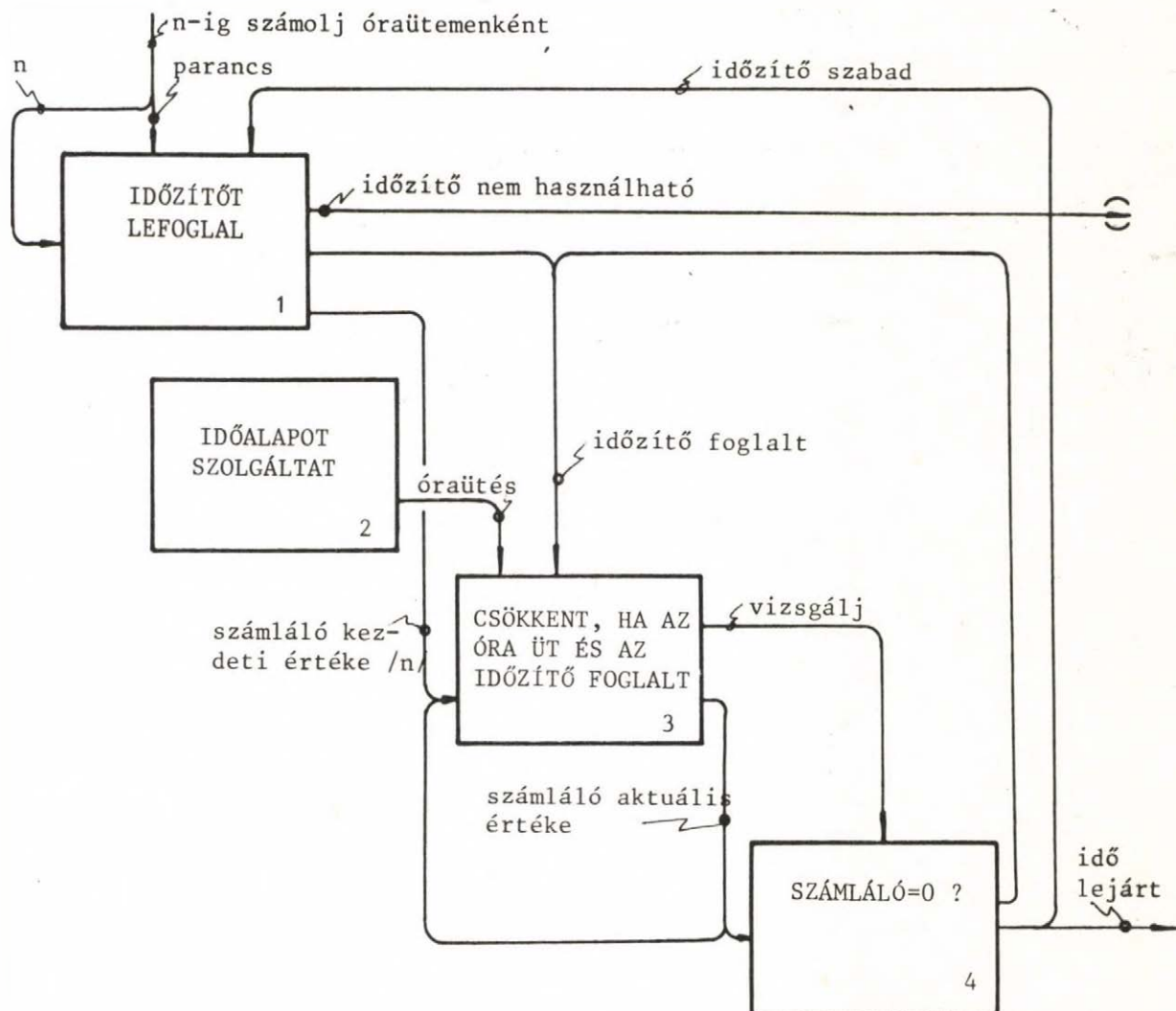


MAKROSZAKASZ

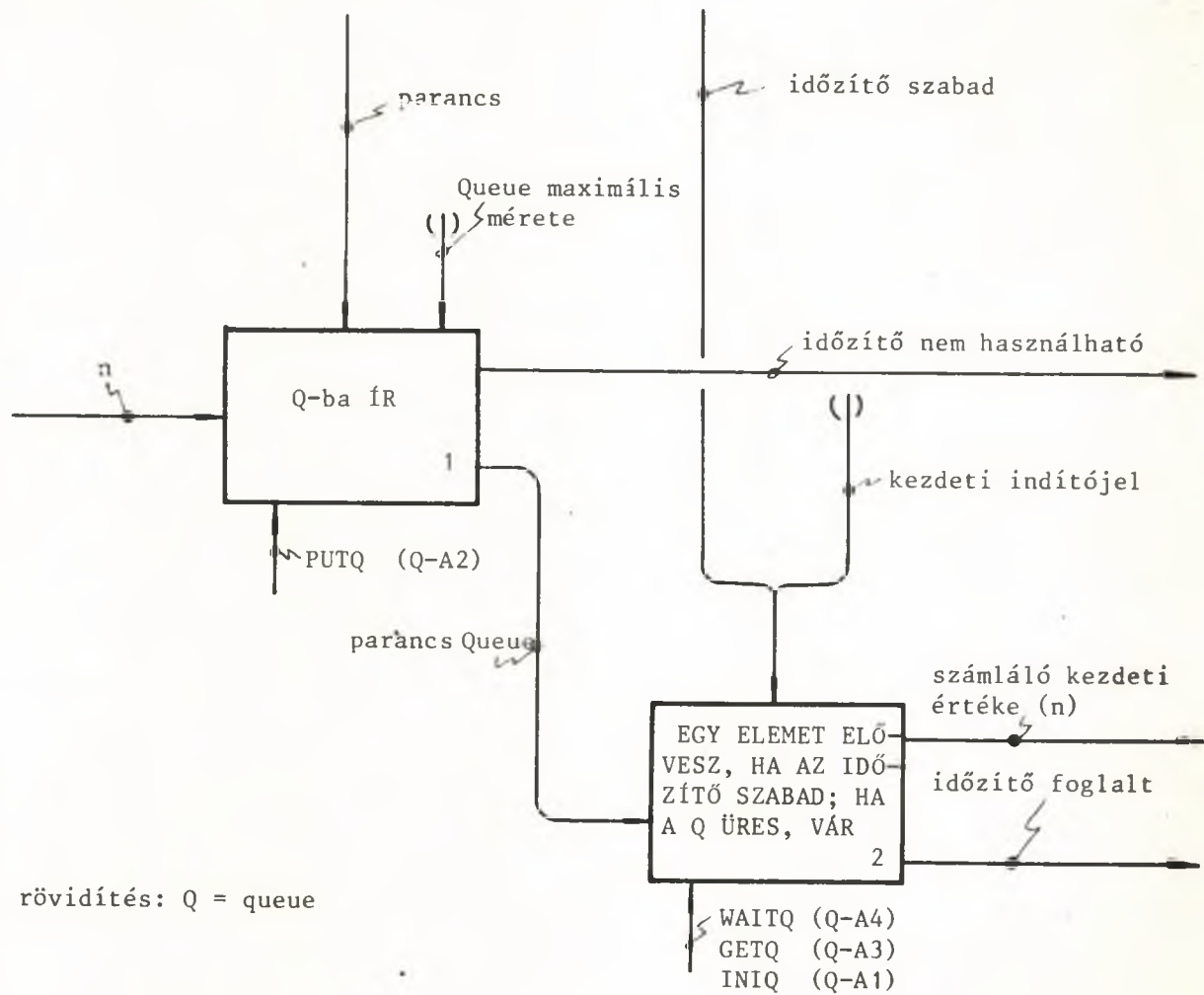
PETRI HÁLÓ

GRAFCE

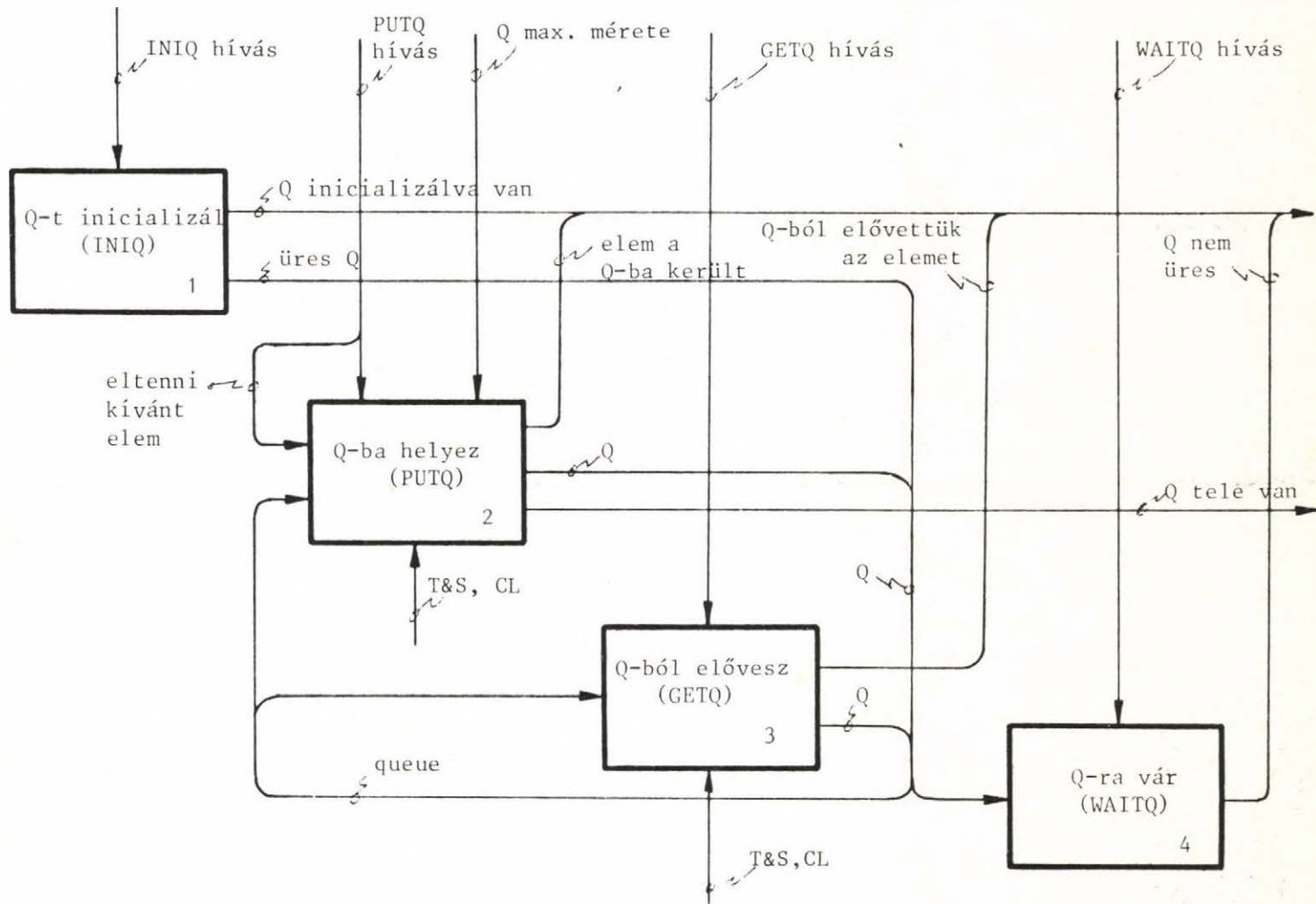
- 2.12 ábra -



Tervezte	Bernus	Cím	t = n időegységet készített				Cég	SZIAKI
Rajzolta								A projekt megnevezése
Gépelte								
Ellenőrizte								
Készítés	1983. június	Munkapéldány		Oldalszám	A szülő változat száma	Változat száma		
Az elavulás időpontja		Ártulás példány			A szülő változat száma	Változat száma		
		Jóváhagyva			Diagram kód		T-A0	



Tervezte	Bernus	Cím	Cég
Rajzolta		időzítőt lefoglal	SZTAKI
Gépelte			A projekt megnevezése
Ellenőrizte			
Keltezés		Munkapéldány	Változátszám
1983 június		Átírdás példány	Változátszám
Az elavulás időpontja		Jóváhagyva	Változátszám
			Diagramkód
			T-A1



rövidítések: - Q = queue
- T&S = test & set

Tervezte	Bernus	Cím	Cég
Rajzolta		QUEUE-kezelő szolgáltatások	A projekt megnevezése
Gépelte		Munkapéldány	A szülő változási szám
Ellenőrizte		Aktuális példány	A szülő változási szám
Keltezés		Jóváhagyva	Változási szám
Az elavulás időpontja			Diagram kód
			Q-A0

hogy

- a Petri modell nagyon világosan és egyértelműen definiálja a vezérlési folyamatot. A vezérlési parancsokat (mint információ típusú dolgot) explicite ábrázolja, azonban az I/O jellegű mennyiségek láttatása csak a szakaszokhoz rendelt predikátumok formájában történik. A Petri-háló alkalmasabb vezérlési struktúra kifejezésére, mint a be- és kimeneti transzformációk megmutatására.
- a Petri-háló a funkciók (átmenetek) vezérlési (gyűjtési) szabályait eleve definiálja, de mint látható, valóságos esetekben e szabályok mellé logikai döntéseket kell írni, melyek az alapszabályokon módosítanak (pl. "Ha=0" a 2.12. ábrán). A SADT modellekben nincs alapszabály, a pontos definíció érdekében (a lebontások folyamán) a funkciókhoz az indítás teljes logikai feltétele hozzátartozik.
- Mivel a SADT modell a sorbandállást csak mint funkcionális lehetőséget kezeli, és nincsenek eleve beépített mechanizmusok, melyek ezt biztosítanák, a SADT modell a sorbandállást megoldó valamely funkció-

ndlis modellt mechanizmusként használja föl. A példában a sorbaállítást elvégző "queue modell" segítségével a SADT explicite jeleníti meg a Petri-hálóknak egy fogalmát, az egy helyen szerepeltethető jelek maximális számát (ld. a 2.15. ábrán "queue maximális mérete").

2.3 Rendszerleíró adatbáziskezelő rendszerek

2.3.1 Az ISDOS rendszer [TEI74]

Az ISDOS rendszer információs rendszerek tervezésére szolgáló számítógépes eszköz, melynek segítségével ilyen rendszerek funkcionális tervei készíthetők el, miközben automatikus analízis és dokumentációs szolgáltatások biztosítják az elkészült végtermék -- a részletes software specifikáció -- jó minőségét, valamint a munka előrehaladtának vezetői áttekintését.

2.3.1.1 Az ISDOS mint a funkcionális modellezés eszköze

Az ISDOS kimondott célja, hogy információs rendszerek tervezé-

sére korlátozza alkalmazási körét, így nem meglepő, hogy kötött fogalomkészlettel dolgozik. Az ISDOS fogalom- és relációkészletét a II./a táblázatban foglaltuk össze [TEI75]. A II./a táblázat a PSL objektumokat, a II./b táblázat a közöttük leírható relációkat sorolja föl. (Kivételesen az eredeti angol nyelvű táblázatot közöljük.)

Az ISDOS rendszer funkciófogalma a SADT-ban megismerttel lényegében azonos. A dolog fogalma jelentősen eltér az általános funkcionális modellektől. Általános dolog-fogalom az ISDOS-ban nem létezik, helyette több olyan fogalmat vezettek be, mely programrendszerek tervezésében szükséges. Egyedül a csoport ("group") fogalma feleltethető meg az általános dolog-fogalomnak, de a be- és kimeneti reláció struktúrájával kapcsolatban nincs semmilyen megkötés,

A funkciók és dolgok (a "process" és a dologi fogalmak valamelyike) közötti relációk szélesebb körét lehet az ISDOS-ban kifejezni. Míg a SADT-ban az ICOM relációk csak egy-egy dolog és funkció közötti bináris relációt írtak le, a PSL nyelv (az ISDOS bemeneti nyelve) magasabbrendű relációkat is megenged. Ilyenekre példa:

INTERFACES OR ORGANIZATIONAL UNITS	INTERFACE (REAL WORLD ENTITY)
=====	
TARGET SYSTEM	
<hr/>	
COLLECTIONS OF INFORMATION	
(EXTERNAL)	INPUT
	OUTPUT
(INTERNAL)	ENTITY
<hr/>	
COLLECTIONS OF INFORMATION INSTANCES	SET
<hr/>	
RELATIONSHIPS AMONG COLLECTIONS OF INFORMATION	RELATION
<hr/>	
DATA DEFINITION	GROUP ELEMENT
<hr/>	
DATA DERIVATION	PROCESS
<hr/>	
SIZE AND VOLUME	SYSTEM PARAMETER INTERVAL
<hr/>	
DYNAMIC BEHAVIOUR	EVENT CONDITION
<hr/>	
PROJECT MANAGEMENT	PROBLEM DEFINER MAILBOX
<hr/>	
PROPERTIES	SYNONYM KEYWORD ATTRIBUTE ATTRIBUTE-VALUE MEMO SOURCE SECURITY
<hr/>	
OTHER	UNDEFINED
<hr/>	

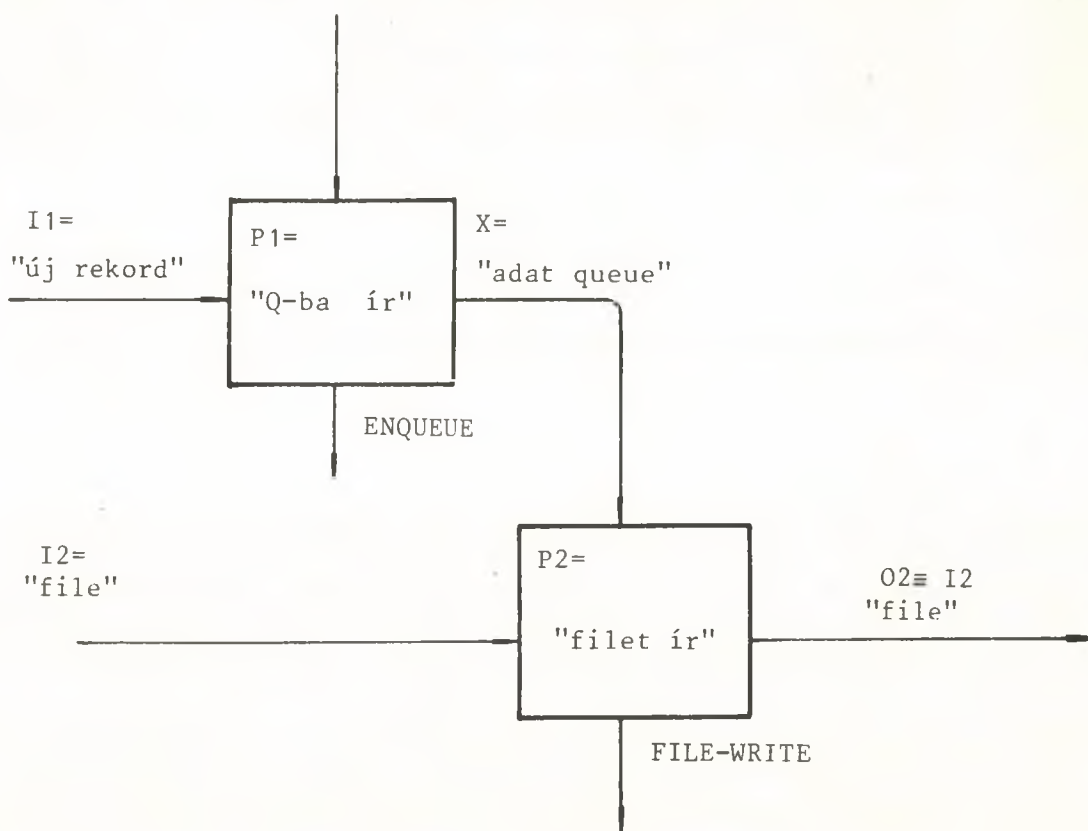
ORGANIZATION OBJECTS		TARGET SYSTEM OBJECTS	PROJECT MANAGEMENT OBJECTS	PROPERTY OBJECTS
ORGANIZATION OBJECTS	SUBPARTS/PART	GENERATES RESPONSIBLE RECEIVES	RESPONSIBLE PROBLEM DEFINER	ATTRIBUTES KEYWORDS SECURITY SEE MEMO SOURCE SYNONYM
TARGET SYSTEM OBJECTS	GENERATED RECEIVED RESPONSIBLE INTERFACE	ASSOCIATED/ASSOCIATED DATA BECOMING /WHEN CARDINALITY CONNECTIVITY CONTAINED/CONSISTS DERIVED/DERIVES GENERATED/GENERATES HAPPENS INCEPTION/INCEPTION CAUSES IDENTIFIED/IDENTIFIES MAINTAINED/MAINTAINS PART/SUBPART RECEIVED/RECEIVES RELATED/RELATES SUBSET/SUBSETS SUBSETTING CRITERIA/SUB SETTING CRITERION TERMINATION/TERMINATION CAUSES TRIGGERED/TRIGGERS UPDATED/UPDATES UTILIZED/UTILIZES VALUES	RESPONSIBLE PROBLEM DEFINER	ATTRIBUTES KEYWORDS SECURITY SEE MEMO SOURCE SYNONYM
PROJECT MANAGEMENT OBJECTS	RESPONSIBLE	RESPONSIBLE	MAILBOX/APPLIES	ATTRIBUTES KEYWORDS SECURITY SEE MEMO SOURCE SYNONYM
PROPERTY OBJECTS	APPLIES	APPLIES	APPLIES	ATTRIBUTES KEYWORDS/APPLIES SECURITY/APPLIES SEE MEMO/APPLIES SOURCE/APPLIES SYNONYM

reláció rendje	reláció leírása PSL nyelven
3	process mér uses mérési adat to generate beavatkozójel;
3	process hozzáfűz uses újrekord to update file;

A magasabbrendű relációk elvben leírhatóak több biner reláció függvényeként, így az az információ, ami a PSL nyelven leírható, az SADT ICOM relációival is kifejezhető.

Mindez csak teljes modellekre mondható el, hiszen a relációalgebrai eszközök használatakor implicit változók is szerepelnek, melyeken keresztül a magasabbrendű reláció előállításához relációk között kapcsolatot teremtünk. Elképzelhető, hogy nem teljes modell esetén a kapcsolati -- az eredményben már nem látható -- változók még nem definiáltak.

A 2.16. ábra az előbbi "update", harmadrendű relációt bontja le részeire. A III.a táblázat a 2.16. ábráról leolvasható biner relációkat sorolja föl, míg a rákövetkező III.b táblázatban azt a relációkifejezést látjuk, melynek eredménye a kívánt update reláció.



P0= "hozzáfűz"

- 2.16 ábra -

!		!
!	Input	<újrekord, Q-ba ír>
!	Output	<adatqueue, Q-ba ír>
!	Input	<file, filelet ír>
!	Control	<adatq, filelet ír>
!	Mechanizmus	<q-ba ír, enq>
!	Mechanizmus	<filelet ír, file write>
!	Output	<file, filelet ír>
!	Része	<Q-ba ír, hozzáfűz>
!	Része	<filelet ír, hozzáfűz>
!		
!		(jelölés:Q=queue)
!		
----- III.a táblázat -----		

!		!
!	update <I1,I2,P0>	<= R <P1,P0> &
!		R <P2,P0> & I <I1,P1> &
!		I <I2,P2> & O <I2,P2> &
!		C <X,P1> & I <X,P2>
!		!
----- III.b táblázat -----		

A relációkifejezésekben P1, P2 és X olyan implicit mennyiségek, melyek a végeredményben nem tükröződnek, tehát ha a lebontás még nem érte el a P1, P2 szintjét, ez a harmadrendű reláció a valóságban még nem generálható.

2.3.1.2 Az ISDOS mint számítógépes eszköz

Az ISDOS programrendszer három alapvető részből áll, u.m.:

- ISDOS adatbázis [RAD80b]
- PSL rendszerleíró nyelv [TEI75]
- PSA analízisprogramok gyűjteménye [BAS75].

Az ISDOS rendszerelemzési szolgáltatásai nem választhatóak el az ábrázolt relációktól. Így a struktúrákra vonatkozó analízis az adott reláció gráfjának tulajdonságaira tett megkötéseket vizsgálja.

Az ISDOS is (a SADT-hoz hasonlóan) fastruktúrát kényszerít a funkciók lebontására, az adatok struktúrájának azonban szélesebb lehetőségeket ad (adattípustól függően fa, irányított körmentes gráf). Az ISDOS adatbázisában tárolt rendszerleírás megjelenítésére szolgálnak a PSA reportgenerátorok. Ezekkel a rendszer úgy használható mint egy általános adatbáziskezelő és lekérdező eszköz [BAS75].

A tisztán funkcionális jellegű kategóriák mellett az ISDOS különféle környezeti és eseménykategóriákat is ismer, melyek pl. a műszaki tervezési szakaszba lépéskor hasznosak lehetnek, továbbá a rendszerterv és az azt előállító személyek közötti kapcsolatot is ábrázolja. Hasonló céllal történik ez, mint a SADT-ban a diagrammok műszaki rajzoknál szokásos feliratozása (diagrammkód, dátum, változatok, változások, szerző neve stb...). Az ISDOS előnyei (automatikus dokumentációtárolás, visszakeresés és analízis) csak információs rendszerek tervezése esetén használhatóak ki maradéktalanul, azonban igen sok szolgáltatása (pl. szelektív visszakeresés, különféle névli-

ták stb.) más rendszerek tervezési eszköztárába való bevonását is indokolja.

Az ISDOS rendszer kiterjesztése általános fogalmi rendszer irányába jelenleg folyik [TEI81]. A további fejlesztés iránya éppen az imént kifogásolt tulajdonságokat küszöböli ki, u.m.:

- A rendszerleíró nyelv (jelen esetben a PSL) kötött fogalomkészletének feloldása. A felhasználási területnek megfelelő fogalmakat meta szinten lehet leírni. A fejlesztési elképzelések szerint (1) ehhez a szinthez csak a rendszergenerálás fokán lehet majd hozzáférfni (tehát felhasználói szinten nem).
- A különböző fogalmi modellek és grafikus reprezentációik közötti kapcsolat létrehozása.

A fogalomkészlet leírása végső soron a fogalmaknak, mint osztályok reprezentánsainak, valamint relációknak megadása. Ide tartozik a relációk struktúrájára tett minden megkötés is.

(1) D. Teichroew szóbeli tájékoztatás, 1980, Budapest

2.3.2 Az SDLA nyelv és rendszer

Az ISDOS projekt sikerétől és felismert korlátaitól hajtattva hazai kutatók is elkezdtek foglalkozni az információs rendszerek számítógéppel segített leírásának és analízisének kérdésével. Ennek eredményeképpen született meg az SDLA (Systems Descriptor and Logical Analyser -- Rendszerleíró és Logikai Analizátor [KNU79], [KNU80], [RAD80a], [DEM82], [KNU82]), mely az ISDOS rendszerhez képest alapvető általánosításokkal élt.

Meta szintű leírással az SDLA felhasználója definiálhatja a leírásban használt relációkat és fogalmakat.

A rendszerleírást tároló adatbázis önmagában is használható mint referenciatípusú relációs adatbázis, ezáltal a rendszerleírás nem entitás-reláció típusú [KNU80], hanem tisztán reláció típusú lett. Ez a tulajdonság megkönnyíti a fogalmak és relációk egyöntetű kezelését.

A relációs adatbázisra épülő lekérdezőnyelv az ISDOS-hoz hasonlóan analízis és dokumentációs célokat szolgáló reportokat hoz létre.

A rendszerleírás kontextusfüggő, miáltal igen tömör és jól

struktúrázható.

A leírás nyelve (mind a meta mind a generált tárgy szinten) nem procedúrális, így procedúrálisan feldolgozhatatlan definíciós struktúrákat is megenged [KNU79].

A relációk egyes gráfelméleti tulajdonságaira meta szinten kikötések tehetők, melyek a leírás adatbázisba vételének feltételévé válnak. Ez a tulajdonság - mint a 4.2.4.2. fejezetben látni fogjuk -- nem minden felhasználási típusban használható ki.

Az SDLA egyfelhasználós (kizárólagos hozzáférésű) adatbázissal rendelkezik, így, ha a rendszerleírásokat tervezőcsoportok állítják elő, a feldolgozást kötegelt módon célszerű megoldani. (Ez az ISDOS rendszerben éppígy igaz).

Az SDLA és hasonló rendszerek (és az ISDOS már említett fejlesztése) a rendszerleírásban és elemzésben új lehetőséget nyitottak meg. Megemlítendő azonban, hogy a meta szint (melyet az ISDOS továbbfejlesztői nem adnak a felhasználó kezébe) az elvégzendő analízisnek mind procedúrális mind relációkifejezésen keresztüli definiálását megkívánná, tehát a felhasználó felé ezen az oldalon is nyitottnak kellene lennie.

Ebben a fejezetben megfogalmazzuk a kutatás alapjául szolgáló koncepciót, és bemutatjuk a megvalósítás felé vezető utat.

3. Célok és munkamódszer

3.1 A funkcionális tervezés módszereire irányuló kutatási koncepció

Az itt röviden ismertetendő koncepció 1976-77 folyamán fogalmazódott meg, azért, hogy ennek alapján végzendő kutatások eredményeképpen létrejöjjön egy számítógéppel támogatott rendszertervezői rendszer, mely alkalmas gépipari integrált anyag- és adatfeldolgozó rendszerek funkcióinak módszeres tervezésére [HAT78].

Az irodalomból ismeretes módszerek a tervezési folyamat más-más részfeladatainak automatizálását tűzték ki célul, közöttük átfedések és felfogáseli különbségek voltak kimutathatóak. (Például két iskola a direkt- és indirekt módszereké [BIE78], melyek szintézisére csak jóval később született javaslat [BER82a].)

A jelen írásban ismertetett kutatómunka alapgondolata, hogy a tervezési folyamat automatizálásának megoldására készíteni kell egy olyan általános modellezési eszközt, mely bármely létező rendszermodellt képes befogadni, és segítségével kiépíthető ezen modellek közötti kapcsolat. Egy ilyen rendszer csak kerete egy olyan tervezői eszköztárnak, melyben nem mindent, nem mindig és nem feltétlenül egyszerre használunk. Ez felel meg leginkább az új rendszerek létrehozásához szükséges modellezési feladatoknak, mivelhogy a tervezett objektum (rendszer) által kielégítendő követelményeket általában nem lehet egyetlen fajta modellben kifejezni (vagy csak nagyon erőszakolt módon).

Akkor mondhatjuk, hogy rendelkezésünkre áll a kívánt rendszer terve, ha

- elkészült mindazon rendszermodell, melyben kifejezett tulajdonságok az összes kiinduló követelményt lefedik,
- az egyes rendszermodellek közötti lehetséges ellentmondásokat kiszűrtük,
- a tervezett rendszer leírása pragmatikus értelemben

teljes (tehát van olyan emberekből, gépekből, mód-
szerekből álló rendszer, melynek számára a modellek
összessége elegendő információt tartalmaz egy eset-
leges szimulációhoz, vagy a tényleges kiviteli ter-
vek megalkotásához -- nem zárva ki a kivitelezésbeli
változatok seregét).

Az általános modellező rendszer felhasználója a megvalósulási
folyamatba beágyazva szeretné eszközeit alkalmazni, tehát a
funkcionális tervezésnek mind bemenete, mind kimenete a kör-
nyező láncszemekhez kell, hogy csatlakozzon.

Egy rendszer "teljes modellje" felfogható úgy, mint egy, a
rendszer állapotváltozói által kifeszített sokdimenziós álla-
pottérben mozgó pont mozgásának leírása. A pont mozgását az
állapotváltozók közötti relációk szabályozzák [BER68]. A pont
mozgására nézve különféle következtetéseket lehet levonni,
általában az állapottér egy tartományát kijelölve az állapot-
vektor számára (ez lehet egy n -dimenziós térgörbe, vagy
valamely többdimenziós felület -- a rendszer meghatározottsá-
gától függően.)

Információtartalmában egy ilyen rendszerleírás minden szem-
pontból elegendő még akkor is, ha több különböző

bázisvektor-halmazzal kifejezett rendszerleírás segítségével adjuk meg a kívánt rendszerjellemzőket. A kívánalmak összehangolására "csupán" megfelelő bázistranszformációkat kellene elvégezni. A gyakorlatban azonban ez a megközelítés igen durva és nem is használható minden esetben. Nem véges, vagy gyakorlatilag végtelen állapotterek esetén ugyanis nem bizonyítható minden esetben két modell ekvivalenciája. Ezért a követelményrendszerekből kiinduló tervezés esetén nincs elvi garancia egy közös explicit rendszermodell létezésére.

Közismert a többtestprobléma, melyet a példa kedvéért így is megfogalmazhatunk: egy sok bolygóból álló együttesnek a mozgása a bolygók tömegközéppontjához rögzített koordinátarendszerben nem fejezhető ki explicit módon, eszerint egy bolygó holdjának mozgása sem. Az adott bolygó és holdja tömegközéppontjának koordinátarendszerében ugyanakkor a hold mozgása minden nehézség nélkül explicite megadható.

Gyakorlatilag használható modelljeinket úgy lehet egy rendszerbe integrálni, ha a közöttük kiépítendő kapcsolati eszköz lehetőséget ad az egyes modellekben használatos fogalmak (entitások, relációk) leírására, majd az egyes modellkapcsolatok kiépítési elvileg dönthetnek a kapcsolat lehetőségeiről és formájáról.

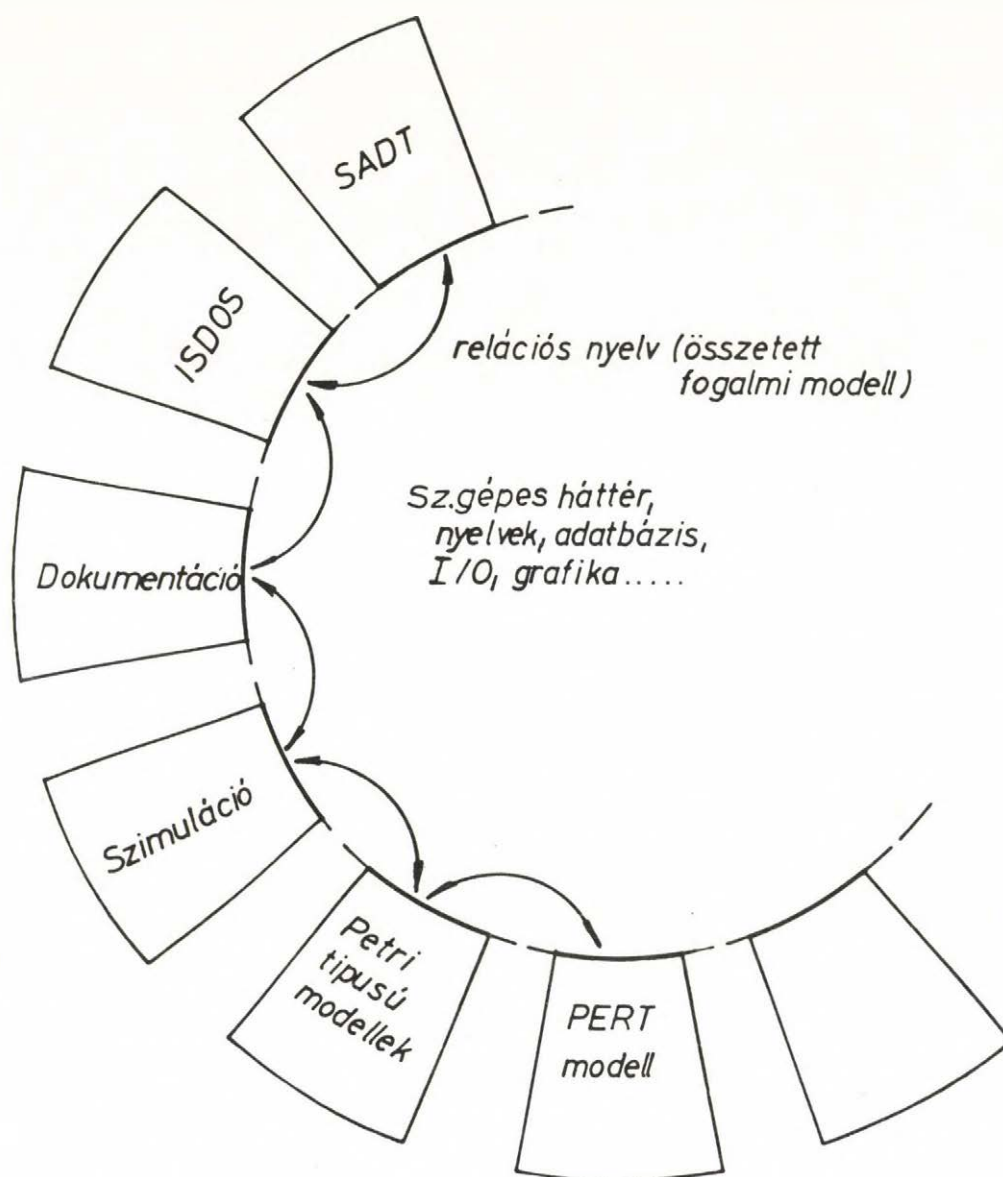
Többféle modellkapcsolat képzelhető el, u.m.

a./ Két azonos célú modell egymásbaalakítása (egy vagy

két irányban). Közös információtartalom kimutatása és kifejezése a másik modellben.

- b./ Két modell közötti ellentmondások kiszűrése, határesetben ellentmondásmentességének bizonyítása.
- c./ Modellek analízise keresett tulajdonságok kimutatására (pl. két modell közös tulajdonságainak megmutatása -- korlátozott ekvivalencia).

A 3.1. ábra sematikusan mutatja, hogyan lehet egy közös mag köré építeni a létező rendszereket. A kapcsolati eszköz egy relációs nyelv. A kutatási feladat kezdetén nem állt rendelkezésre felhasználható nyelv, sem pedig adatbázis -- ennek kiépítése pedig nagyon megterhelte volna a kutatási projektet. Ezért úgy döntöttünk, hogy egy olyan egyszerű és könnyen megvalósítható relációs felületet definiálunk, mely a későbbiekben (ha általános adatbázis születik) biztosan létrehozható annak segítségével is. A választott felület egy asszociatív adatbázis, melynek rövid ismertetése az I. függelékben olvasható. Bármely modellt kifejezve ezen a relációs nyelven, formailag egységes lesz a rendszertervezői rendszer információ tartalma. Ezt a tartalmat azután az egyes modellkapcsolatokat megvalósító algoritmusok egységes módon használhatják.



- 3.1 ábra -

3.2 Stratégia és munkamódszer a rendszertervezői rendszer egyes elemeinek megvalósítására

Az általános rendszertervezői rendszer szolgáltatásainak kiépítésére az alábbi lépéseket állítottuk föl:

- 3.2.1. Követelményrendszerek struktúrált megfogalmazása, funkcionális fölépítmény meghatározása.
- 3.2.2. Az 1. szakasz automatikus dokumentációja és támogatása számítógépes szerkesztői eszközökkel.
- 3.2.3. A funkcionális felépítmény tárolása adatbázisban, visszakeresési, lekérdezési, automatikus szöveges dokumentálási lehetőség.
- 3.2.4. Szemantikai analízis egyes lépéseinek automatizálása a funkcionális modellek vizsgálatára.
- 3.2.5. Kapcsolat kiépítése szimulációs és más dinamikus modellekkel (Petri, PERT...)
- 3.2.6. Különböző megvalósítási stratégiák tervezése és au-

tomatikus támogatása.

Amint látható, a rendszerépítés eme stratégiája nem támaszkodik egyetlen üdvöztető módszerre sem. Ehelyett a rendszermegvalósulási folyamat egyes fázisaiban használt többféle módszer közös keretévé kíván válni. Ez maga után vonja azt is, hogy a feladatkijelölésnek minduntalan kétféle válaszüttja van: a rendszertervezői rendszert bővíteni a permisszivitás irányába (horizontálisán) -- illetve a megvalósulási (innovációs) folyamat mind több szakaszának eszközeit bevonni a támogatottak körébe (vertikális bővítmények).

A 4.1.2 fejezetben ismertetett rendszerkapcsolat nem egyszerűen két rendszer összekötése, hanem a vázolt keretrendszer első két tagja.

Ad 3.2.1. A követelményrendszer struktúrált megfogalmazásának eszközeül egy konkrét módszert szükséges kiválasztani, mely (figyelembe véve a szakterületi és általános igényeket is) az alábbi tulajdonságokkal kell, hogy rendelkezzen:

- funkcionális modellezésre alkalmas legyen, tehát válassza szét a feladatokat és megvalósításukat.

Segítségével egyértelmű leírást lehessen készíteni nagy rendszerek funkcióról és csatlakozási felületeiről.

- Figyelembe véve, hogy igen bonyolult rendszerek modellezéséről van szó (melyek ráadásul több szakterület együttes erőfeszítéseként jönnek létre), legyen képes tetszőleges szakterület fogalmait modellezni, és tudja követni a modellezendő rendszer méretének növekedését, megőrizve az áttekinthetőség tulajdonságát. Álljanak rendelkezésre a modellek vizsgálatára alkalmas általános módszerek és elvek.

- Legyen a módszer praktikus, tehát az elméleti alkalmasság kezelhetőséggel, jól rendszerezhetőséggel párosuljon. A gyakorlati alkalmasságot javító tényező, ha a módszer nem csak számítógépes támogatással használható.

- Legyen a modellezési eljárás közérthető. Ez az egyik sarkallatos pont, hiszen igen sokféle szakember vesz részt a rendszerterv megalkotásában, akik a modell nyelvezetét gondolatközlésre, megállapodások egyértelmű rögzítésére használják. A nyelvnek így

nem szabad valamely kitűntetett szakterület fogalmának egyedüli előtérbe állítására épülnie -- minden résztvevőtől nagyjából azonos absztrakciós szintet követeljen meg. A választott módszer alkalmasságát gyakorlati példán keresztül is igazolni kell.

Amellett, hogy szükség van egy itt vázolt célú rendszertervezői eszköztár folyamatos kiépítésére, specifikus célok érdekében ki kell dolgozni rendszerezett lépések sorát, melyet pl. egy gyártórendszer tervezője az eszközök támogatását is felhasználva követhet. Leszűkítve a feladatot a funkcionális tervezés területére, szintén szükség volt olyan módszertanra, mely eligazítást ad a funkcionális tervezés menetéről (a modellkészítés lépései és szabályai, gondolkodásmódja, minőségi kritériumai stb.)

Ilyen módszertan kidolgozására az alábbi stratégia látszott alkalmasnak:

- a funkcionális tervezési eljárás kipróbálása kisméretű feladaton [BER79a],

- nagyméretű valós rendszer analízise kísérleti célból [BER80],
- módszertan kidolgozása [BER81c], [BER81d],
- módszertan alkalmazása valódi gyakorlati feladatra [BER81e], [BER82d],

Ad 3.2.2. Létre kellett hozni a választott módszer számítógépes támogatásának eszközeit, figyelemmel a későbbi rendszerbekapcsolhatóság követelményére is. A számítógépes támogatás fő célja a modellek

- interaktív szerkesztése
- dokumentálása, tárolása,
visszakeresése, naprakészen tartása
- formai vizsgálata

volt. A rendszert ki kellett próbálni valós méretű kísérleti feladaton.

Ad 3.2.3. A funkcionális modellek analízisének támogatására a létező adatbáziskezelő rendszerek közül olyat kellett választani, mely alkalmas volt a választott modellezési módszerrel készített leírások tárolására, szelektív lekérdezési lehetősége volt és szöveges dokumentációt is tárolni lehetett segítségével. Ennek ellenőrzését szintén el kellett végezni gyakorlati példán [PAL78].

Ad 3.2.4. A modellek analízisének automatizálására -- tekintetbe véve, hogy a funkcionális modellek teljes információtartalma relációkifejezések segítségével leírható -- olyan rendszert kellett választani, mely az összes készített modell ilyen jellegű leírását tárolni képes (pl. relációs adatbázis), továbbá az analízishez szükséges szolgáltatásokkal rendelkezik (pl. relációs lekérdezőnyelv).

A szemantikai analízis elvégzésének előfeltétele volt a funkcionális modellezés szemantikai szabályainak kidolgozása, és a választott modellezési módszer formalizálása.

Ad 3.2.5. Kapcsolat más, dinamikus modellekkel:

Ad 3.2.5/1. Az eddigi lépések az innovációs folyamat egyes lépéseinek lefedésére törve logikus egymásutánban következtek. A koncepció szerint a rendszertervezői rendszernek a tervezői munka erősen individuális stílusait és eszközeit kell összeegyeztetnie a végeredmény egységességének, konzisztenciájának követelményével. Hasonló, vagy átfedő feladatkörű eszközök beépítése esetén tehát az átfedő információtartalmú modellek közötti kapcsolatot is létre kell hozni. Ilyen lehetőség pl. a Petri-hálós tervezés valamely változatának beépítése a rendszertervezői rendszerbe. Ezek a munkák a rendszert horizontálisan bővítik.

Ad 3.2.5/2. Az innovációs folyamat következő láncszemeihez való csatlakozás a rendszertervezői rendszer vertikumban jelenthet előrehaladást, pl. az ellenőrzött funkcionális modellek valamely szakterületi megvalósításának irányában.

Ad 3.2.5/3. Az Ad 3.2.5/1 és /2-vel szorosan összefügg a szimulációs modellekkel kiépíthető kapcsolat. Egy Petri-háló már szimulációs vizsgálatokra is

alkalmas, másrészt egy számítógépprogram (mint megvalósítás) egy másfajta megvalósítás szimulációs modelljeként fogható föl. (A szűkebb értelemben vett szimuláció abban különbözik a megvalósítástól, hogy közege homogén.) A szimulációnál általánosságban, a számítógépprogramozásnál pedig speciálisan, nemcsak a működtethető modell (a "megvalósítás"), hanem a működtető környezet (szimulációs környezet, teszt-környezet) megtervezésére is használhatónak kell lennie a kidolgozott rendszernek.

Ebben a dolgozatban a 3.2.1, 3.2.2., 3.2.3. és részben a 3.2.4. pontban ismertetett célkitűzések megoldását és az annak kapcsán született tudományos eredményeket mutatjuk be.

3.3. További lehetőségek

A 3.2. Ad 3.2.5-ben felsoroltak megvalósításához kezdeti lépéseket éppen az irodalmi összefoglaló összehasonlító elemzése nyújthat az egyes Petri-háló típusok és a tisztán funkcionális modellek viszonyának bemutatásával. Az Ad 3.2.5/2 és Ad 3.2.5/3 feladatok megoldásához az elméleti vizsgálatok mellett gyakorlati (kísérleti) tapasztalatok állnak rendelkezésre

[BER79a], [BER82b], [BER83a,b,c], [KOV83]. Ezek elsősorban olyan feladatmegoldások, melyeknél a tisztán funkcionális tervezést kézi úton hajtottuk végre, majd az eredményül kapott modelleket kézi úton ültettük át számítógépi programnyelvre.

A számítógépprogramok esetében a direkt megvalósítás stratégiája az általános (többdiszciplínájú) esetben sokkalta egyszerűbb, lévén az építkezés primitívkészlete kötött. Elméleti vizsgálatokat lehetne abban az irányban folytatni, mely kimutatná, hogyan lehet az általános modellek programrealizációra kijelölt részeit teljesen, vagy részben automatikus eszközökkel a programtervezési módszerek speciális funkcionális modelljeire átfordítani.

A másik út az eddig követett kézi módszer támogatása automatizmusok segítségével. A kapcsolat a realizációval gyártórendszerek esetén minimálisan azt igényli, hogy ne csak számítógépi programok, hanem gépészeti konstrukciók és elektronikus berendezések megvalósítói számára is megfelelő feladatmeghatározást lehessen előállítani. A folyamat teljes automatizálása nem cél, bár vannak területek, ahol ennek elvi akadálya nincs (pl. a GRAFCET-hálóak megvalósítása PLC programokkal [ADE80], vagy mikroprogramozott vezérlések tervezése Petri-hálókkal [BAN80]).

További lehetőség, ha a rendszertervezői rendszert nemcsak egy tárgyrendszernek, mint műszaki alkotásnak, hanem egy kivitelezési feladatnak a megtervezésére használjuk (esetleg éppen a tárgyrendszer kivitelezésére). A kivitelezési funkciók modellbe foglalására alkalmas módszerek és a PERT vagy CPM hálók közötti kapcsolat fölépítését kívánná meg ez a fejlesztési irány. Külön előny várható ettől a módszertől, mivel várhatóan struktúrált PERT hálók előállítását segítené elő, ami nagy és áttekinthetetlen hálótervek létrejötte ellen hatna.

A módszertan kidolgozásában előrelépést jelenthetne, ha a funkcionális tervezés bemenetét képező követelményrendszerek kialakítására alkalmas innovációs módszertanokkal építenénk ki a kapcsolatot [LAD81], [LAD82], továbbá a funkcionális tervezés és az értékelemzés közötti információcserére is fölkészítenénk a rendszertervezői rendszert.

Ez a fejezet két lényeges részre tagozódik. Az első (4.1) rész a funkcionális modellezési módszer számítógépes támogatásának és a létrehozott rendszer gyakorlati kipróbálásának eredményeiről szól. Ez a fejezet számol be a rendszerfejlesztői rendszer koncepciójának megvalósítására kialakított stratégia első három lépésének megtételéről.

A második (4.2) rész a negyedik stratégiai lépést taglalja: beszámol a választott funkcionális notáció formalizálása, valamint szemantikai analízise terén elért saját eredményekről és a szemantikai analízisre alkalmas rendszer kiválasztásáról.

4. SAJÁT EREDMÉNYEK A FUNKCIONÁLIS RENDSZERTERVEZÉS TERÜLETÉN

4.1 Funkcionális modellek adatbázisának definiálása és kialakítása

Az irodalmi áttekintésben összefoglalóan ismertetett módszerek és eszközök, valamint a 3.2 pontban meghatározott követelmények alábbi összefoglalója bemutatja, hogy a gyártórendszerek funkcionális tervezésére, továbbá a folyamat számítógépi támogatására mely rendszer milyen mértékben alkalmas (IV. táblázat).

fogalomkészlet:	SADT :	kötött, általános
	Petri:	kötött, általános
	ISDOS:	kötött, információs rend- szerek leírására jó
	SDLA :	kötetlen, általános
kategóriák:	SADT :	funkció, dolog (információ, anyag, állapot stb...)
	Petri:	átmenet, állapot, vezérlési folyamat
	ISDOS:	process, esemény és adatkate- goriák, management információ + egyéb attribútumok.
	SDLA :	tetszőleges
relációk:	SADT :	be- és kimenet, vezérlőbemenet és mechanizmus
	Petri:	be- és kimenet
	ISDOS:	be- és kimenet, mechanizmus, vezérlés
	SDLA :	tetszőleges
struktúrálás:	SADT :	funkciók és dolgok struktúrált leírása
	Petri-típusok:	
	Petri :	nem struktúrált
	GRAFCET:	funkcióstruktúrát írja le
	AP-net :	funkció- és állapotstruktúra
	ISDOS:	funkciók és dolgok struktúrált leírása
	SDLA :	tetszőleges, definiálható
modellezés nyelve:	SADT :	grafikus notáció
	Petri:	grafikus notáció
	ISDOS:	PSL szöveges leírónyelv
	SDLA :	szöveges meta és tárgynyelv
analízis eszközei:	SADT :	kézi + módszertan az előállít- ásra
	Petri:	kézi funkcionális analízis + szimuláció vagy mátrixalgebrai algoritmusok
	ISDOS:	adatbázis + lekérdezőnyelv + reportgenerátorok (algorit- musok)
	SDLA :	adatbázis + lekérdezőnyelv + reportgenerátorok (algorit- musok)
rendelkezésre állás (a választás időpontjában /1976-77/):		
	SADT :	publikációk
	Petri :	publikációk
	ISDOS:	implementált verzió több hazai gépen
	A többi ma (1984-ben) ismert módszer még nem létezett	

Néhány modellezési eszköz és módszer jellemzői

Ez hivatott bemutatni azokat az indokokat, melyek a koncepcióban vázolt rendszertervezői rendszer kialakításának első lépésévé tették a SADT módszer számítógépi grafikus támogatását, majd összekapcsolását az ISDOS rendszerrel.

A SADT és az ISDOS rendszer összekapcsolásából kialakítható rendszertervezői rendszer tulajdonságai a IV. táblázatban kiemelt írásmóddal szerepelnek.

4.1.1 A funkcionális modellezés módszerei közötti választás indoklása

A SADT mint modellezési eszköz előnyei komplex gépipari rendszerek tervezésénél az alábbi tulajdonságok:

- Interdiszciplinárisan alkalmazható fogalomkészlet, egyenértékű módon képes az anyag és információfolyamot ábrázolni. Nincs eleve kitüntetett dologi fogalom (az adat és vezérlésfolyamon belül sem).
- Grafikus modellekkel dolgozik, ami általában is, de a gépiparban különösen könnyíti a megértést, lévén a rajz a bevett gondolatközlési forma.
- A módszer a gyakorlati alkalmazásokat tekintve jó referenciákkal rendelkező, nagy gépipari projektekben sikerrel alkalmazták [AF 78].

- további előnyök a struktúrált modellezési módszer, valamint az, hogy alkalmazni lehet számítógép nélkül is.

Néhány hátrányos tulajdonságról az irodalmi összefoglaló már ejtett szót.

Az ISDOS rendszer előnyei:

- jó dokumentációs készség: a tervezett rendszerről szóló minden információ számítógépi adatbázisba kerül, és onnan lekérdezhető.
- Az ISDOS több számítógépen is rendelkezésre állt.

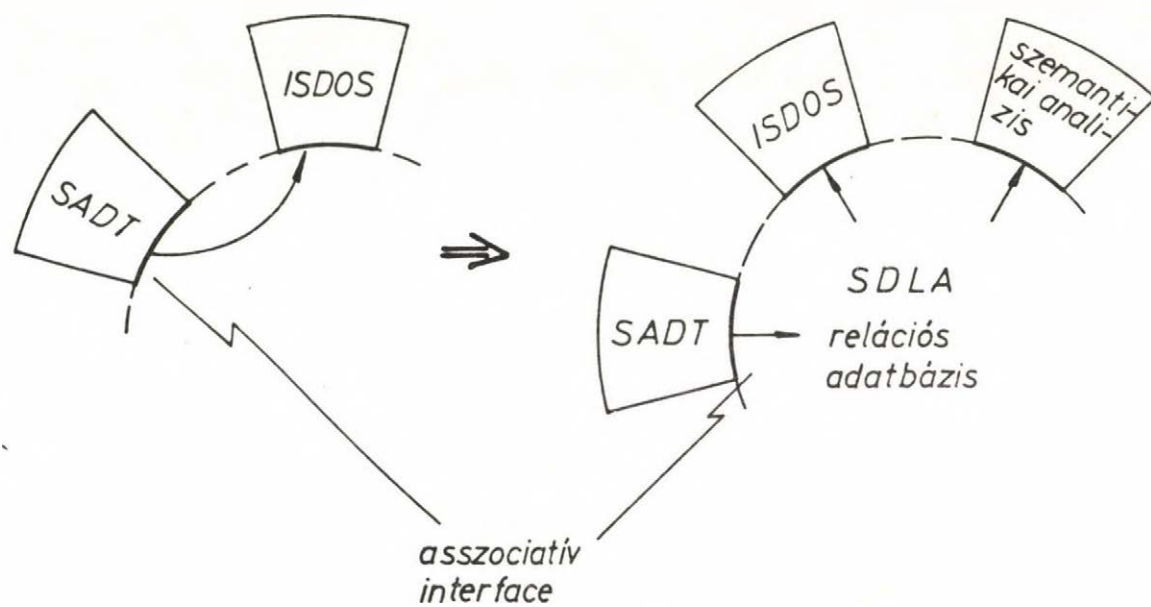
Az irodalmi összefoglalónál általánosságban megállapított korlátozások az adott esetben azt eredményezik, hogy gyártórendszerek funkcionális tervét PSL-ben leírva a fogalmak összerendelését kell elvégezni. Ez a megjelenési formán valamelyest ront. Az ISDOS emellett viszonylag nagy számítógépigényű programrendszer.

A Petri-háló -- bár az elméleti megalapozottságuk matematikai oldalról az előzőekénél erősebb -- több olyan nehézséget

tartogatnak a gyakorlati felhasználásnál, melyek első rendszerelemkénti választásukat nem indokolták. Ezek az okok:

- Nem struktúráltak, így a modell kialakítási folyamatához nem adnak segítséget. Ez a modellezett feladatméretnél (több száz, vagy ezer funkció és dolog) már igen lényeges szempont. A Petri háló -- gyártórendszerek tervezésében -- elsősorban bizonyos problémátípusoknál alkalmazható előnyösen (pl. egyes termelésirányítási algoritmusok modellezése és analízisa).
- A Petri hálók az anyag- és információfolyamnak egy speciális oldalát hangsúlyozzák (állapotátmenetek folyamata és azok összefüggései).

Az itt felsorakoztatott érvek és tények utólag is alátámasztják annak az elvnek a helyességét, mely szerint több különféle eszközt be kell fogadnia egy rendszertervezői rendszernek. Ez az, amitől a két első rendszerelem összekapcsolása új módszerek megjelenésétől nem elavulttá válik, hanem ellenkezőleg: új módszerek gazdagítják a lehetséges, és adott pillanatban követhető utak számát. Pl az SDLA mint relációs adatbázis és rendszerleíró nyelv éppen a 3.1. ábrán üresen hagyott, csak



- 4.1 ábra -

csatlakozó felületében definiált rendszerkomponens helyét töltheti be. Ezáltal megszüntethető a kapcsolat közvetlen jellege (t.i. tárolás is közbeiktatható) -- ld. 4.1. ábra.

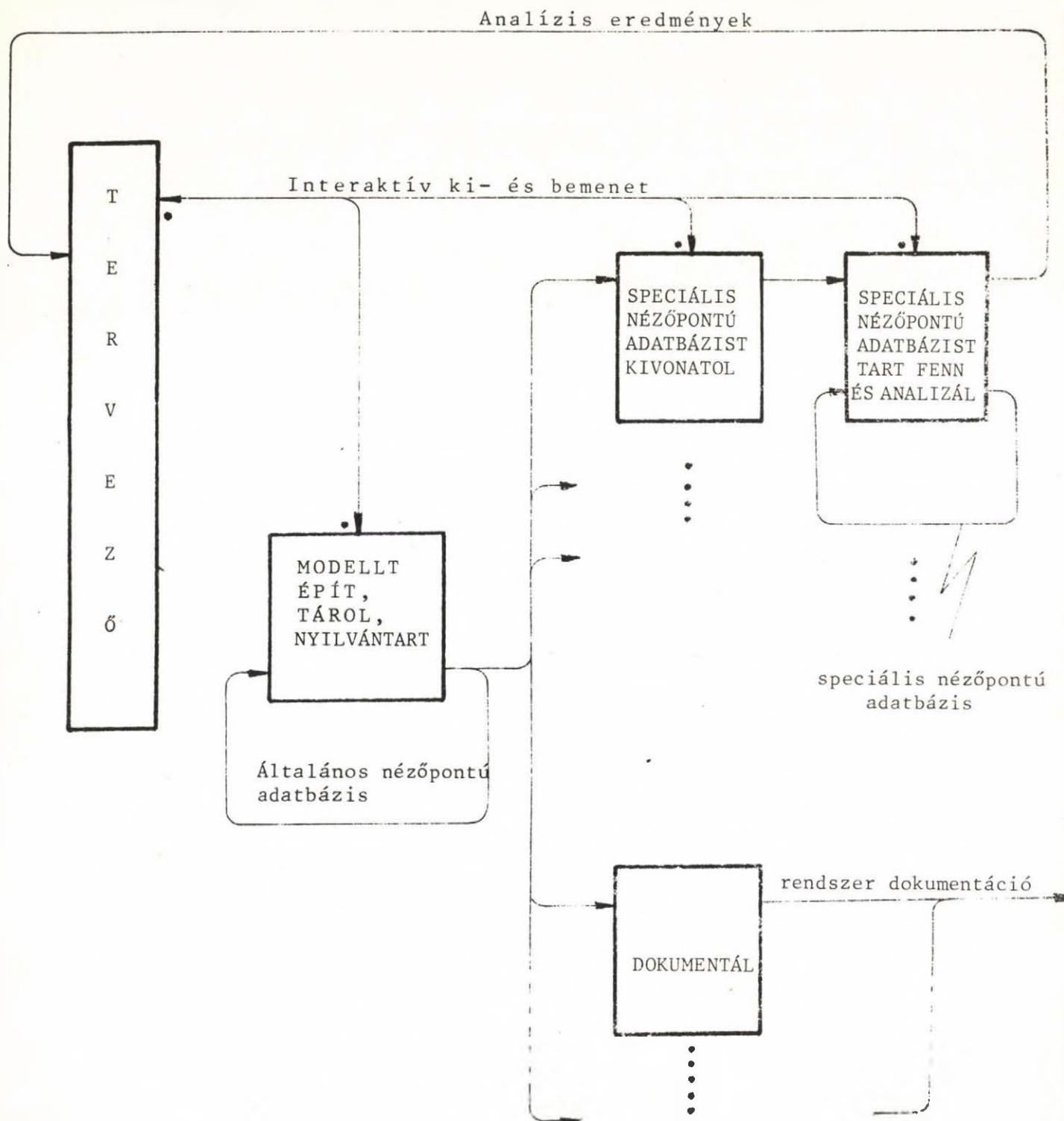
4.1.2. A SADT és az ISDOS rendszer összekapcsolása

Ebben a fejezetben ismertetjük azt a számítógépes programrendszert, melyet a rendszertervezői rendszer első két elemeként választott SADT és ISDOS összekapcsolása révén hoztunk létre.

4.1.2.1. A SADT - ISDOS rendszer

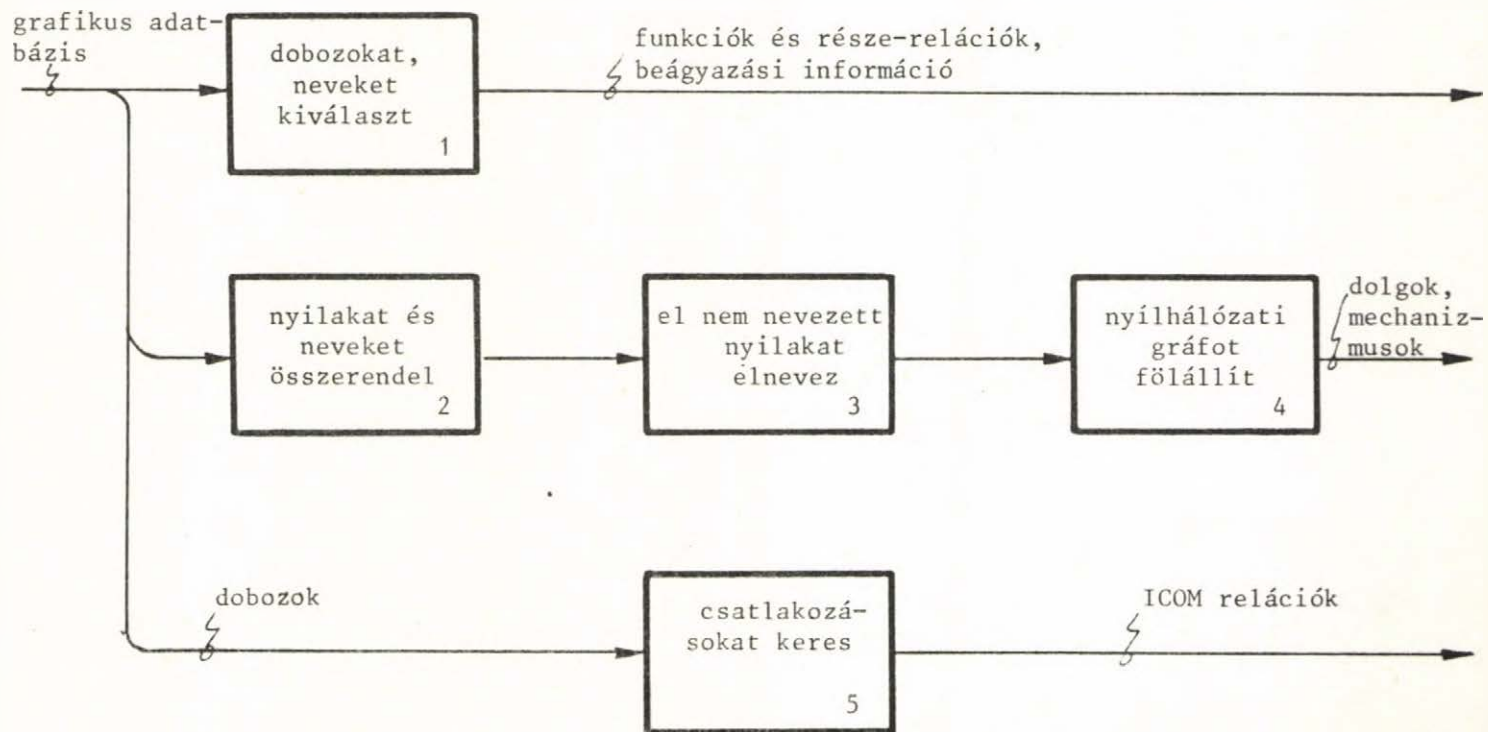
A rendszer feladatát a 4.2. ábrán vázoltuk. A SADT modellek grafikus nyilvántartására, interaktív szerkesztésére szolgáló rendszert 1978-79 folyamán meg is valósítottuk. A felhasznált eszközök: TPA'70 kisszámítógép, GD'71 grafikus megjelenítő, CALCOMP rajzgép és néhány hagyományos számítógépperiféria. Erről számol be a többi között [BER79b].

A SADT grafikus rendszert az asszociatív interfacehez csatlakoztatandó, meg kellett fogalmazni a SADT modellek információtartalmát bináris relációk segítségével. A grafikus adatstruktúrát a 4.3. ábra szerinti funkciók alakítják át



- 4.2 ábra -

Tervezte	Bernus	Cím	Cég
Rajzolta		grafikus adatbázis tartalmának kivonatolása	A projekt megnevezése
Gépelte			
Ellenőrizte			
Képezés			
Az elavulás időpontja		Munkapéldány	A szülő változat szám
		Aktuális példány	Változat szám
		Jövőbogyva	Változás szám
			Diagram kód
			A0

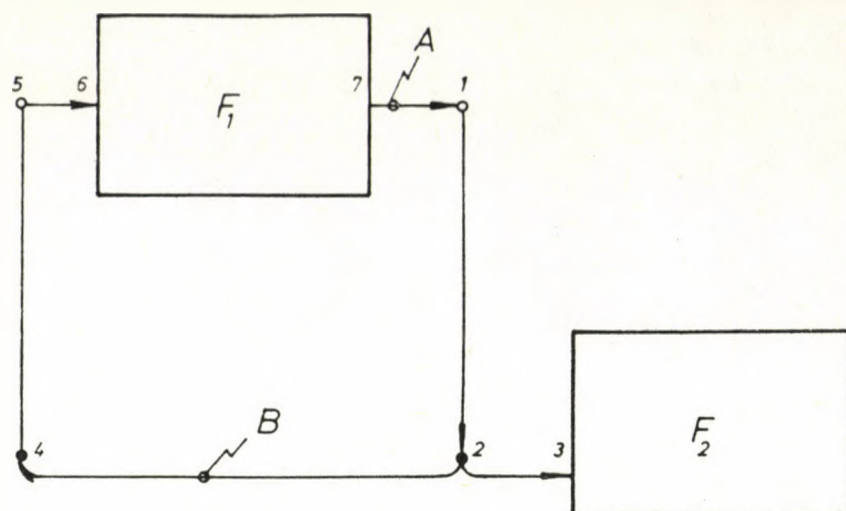


megjegyzés: A3 végzi a névkiterjesztést

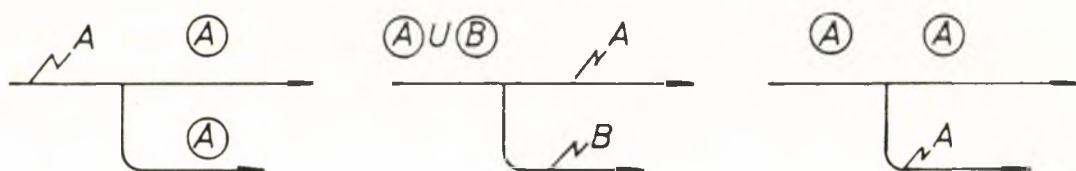
relációkifejezésekké.

Ezek a funkciók egyszerűen algoritmizálhatóak, külön említést csak a "névkiterjesztés" igényel. A 4.4. ábra mutatja, hogy a grafikus modellekben a nyílhálózat egyes ágaira írt nevek a szemlélő (a grafikus modell rajzolója, vagy emberi interpretátora) számára egyértelműen kell, hogy jelöljék az egyes nyilak által reprezentált dologi- (vagy funkció-) osztály nevét. Bizonyos esetekben (pl. 4→5, 5→6 ágak) nyilvánvaló, hogy a "B" név a nyíl irányában kiterjeszthető, tehát F1 funkciónak "B" lesz a bemenete. Az 1→2→3 ágaknál az "A" név választása tűnik logikusnak. Ezekben az esetekben éppen az a korrekt, ami logikusnak tűnik, mivel a névkiterjesztési algoritmus éppen az emberi szemlélő magatartását kell, hogy utánozza. A [BER78]-ben összefoglalt névkiterjesztési algoritmus természetesen pontosan definiálja, hogy az értelmezés milyen módon történjen, ez azonban nem volna elegendő, ha az algoritmus a valódi esetek túlnyomó többségében nem az emberi reakciót reprodukálná. Ezért bizonyos körülmények között, amikor az algoritmus kétséges esethez ér, -- mégna logikusan alkalmazható is volna az adott esetre -- a kezelőt a bizonytalanságról tájékoztatja.

Néhány alapesetet a 4.5 ábra mutat be (bekarikázva az automa-



- 4.4 ábra -



- 4.5 ábra -

tikus névkiterjesztés által választott nevet):

4.1.2.2 A SADT modellek alapvető információtartalma

A SADT modellek grafikus nyelvének és az általa leírt információknak megfeleltetése:

Aktigrammok (funkciók lebontása)

- Minden téglalap egy funkciót jelöl (F)
- Minden nyíl egy dolgot jelöl (D)
- A lebontott funkció (Fx) és részei (Fx1, Fx2, ...) között a "része" reláció áll fenn $\langle Fx, Rf, Fx1 \rangle$
Ha egy nyíl (D) belép a lebontott Fx funkcióba, az
 - $\langle Fx, I, D \rangle$ bemeneti reláció vagy az
 - $\langle Fx, C, D \rangle$ vezérlő (kontroll) reláció áll fenn.
- Ha egy nyíl elhagy egy funkciót ábrázoló téglalapot,
 - $\langle Fx, O, D \rangle$ kimeneti reláció áll fenn.
- Ha egy (E) nyíl alulról lép be a funkciót ábrázoló téglalapba, akkor annak jelentése: a funkció végrehajtásához szükség van az adott E eszközre (mint mechanizmusra), így az
 - $\langle Fx, M, E \rangle$ mechanizmusreláció áll fenn (itt E egy általános célú mechanizmusfunkció valamely megvalósítását jelöli).

 Datagrammok (dolgot lebontása)

- Minden téglalap egy dolgot jelöl (D)
 - Minden nyíl egy funkciót jelöl (F)
 - A lebontott dolog (Dx) és részei (Dx1, Dx2, ...) között a "része" reláció áll fenn $\langle Dx, Rd, Dx1 \rangle \dots$.
Ha egy nyíl (F) belép a lebontott Dx dologba, az $\langle Dx, I, F \rangle$ bemeneti reláció vagy az $\langle Dx, C, F \rangle$ vezérlő (kontroll) reláció áll fenn.
 - Ha egy nyíl elhagy egy dolgot ábrázoló téglalapot, $\langle Dx, O, F \rangle$ kimeneti reláció áll fenn.
 - Ha egy (E) nyíl alulról lép be a dolgot ábrázoló téglalapba, akkor annak jelentése: a dolog megvalósításához szükség van az adott E eszközre (mint mechanizmusra), így a $\langle Dx, M, E \rangle$ mechanizmusreláció áll fenn (itt E egy általános célú mechanizmusstruktúra megvalósítását jelöli).
-

Amint azt a funkcionális modellek formalizálásánál részletesebb tárgyalásban látni fogjuk, létezik ezen kívül több más reláció is, melyeket azonban az ISDOS rendszerben nem lehet leírni és jelentőségük a több modell közötti ellentmondásmentesség vizsgálatánál van. A mechanizmusfogalom irodalmi tisztázatlansága miatt első lépésben csak a számítógépprogramozásban szubrutin- (vagy makro-) hívásként ismert fogalom általánosításaként értelmeztük a mechanizmusrelációt, a tágabb és egyértelmű definíciót a 4.2.1 fejezetben adjuk meg.

A fenti bindris relációkat a SADT diagrammok grafikus modelljéből kialakító program ugyancsak része a [BER79b]-ben ismer-

tetett rendszerünknek.

A SADT-beli fogalmak és relációk valamint az ISDOS PSL nyelvének fogalmai és relációi között az alábbi megfeleltetést lehet létrehozni:

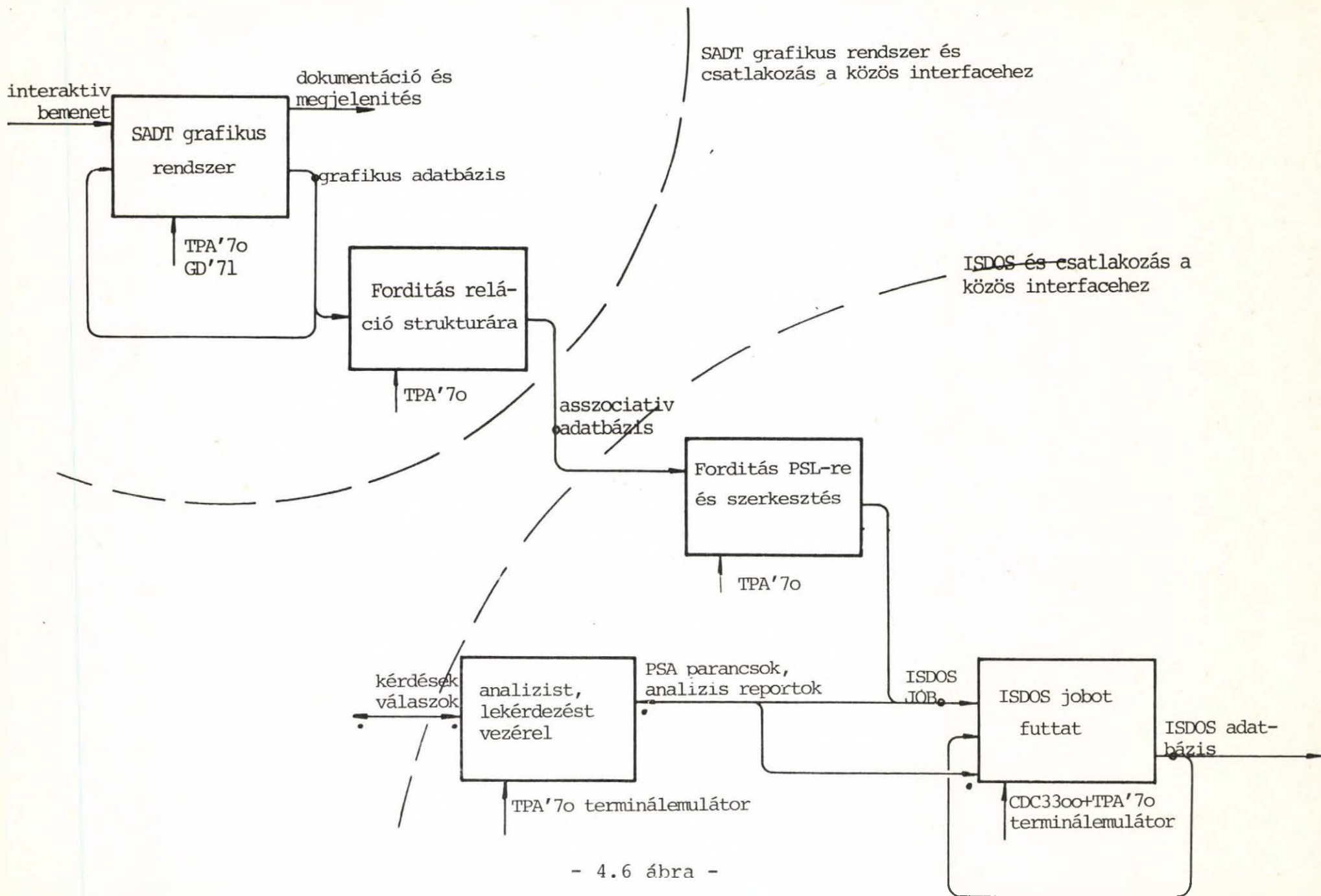
! SADT	! ISDOS/PSL	!
! ----	! -----	!
! funkció	! process	!
! dolog	! group	!
! <...része...>	! <...part-of...>	!
! <...input...>	! <...uses...>	!
! <...output...>	! <...derives...>	!
! <...kontroll...>	! <...uses...>	!
! <...mechanizmus...>	! <...utilizes...>	!
! funkció definíciója	!	!
! aktigrammon	! keyword: explicit	!
! datagrammon	! implicit	!
! dolog definíciója	!	!
! aktigrammon	! keyword: implicit	!
! datagrammon	! explicit	!

A modellek PSL nyelvű kifejezését az asszociatív struktúrákra alkalmazott operátorok (I. függelék) segítségével (azok algoritmikus formájában) már csak programozási rutinfeladat volt elvégezni.

A gyakorlati megvalósításban a kisszámitógépen rendelkezésünkre álló távoli kötegelt adatfeldolgozási kapcsolatot használtuk föl az ISDOS rendszer elérésére, melynek egy CDC3300-as gépen futó változata a SADT-PSL fordítás eredményeképpen

kapott feladatot (JOB-ot) dolgozta föl. Az ISDOS rendszer által fölépített adatbázis lekérdezése, módosítása így a kisszámítógépen keresztül történhetett (erre a TPA'70-es számítógépen a CDC 3300-as számítógép 200-as felhasználói terminálját emuláló program adott lehetőséget) [BER79b]. A 4.6. ábra mutatja be a kapcsolat lényegét. Az ismertetett rendszer volt a világon az első (1) , melyen grafikusan lehetett helyességre ellenőrzött SADT diagrammokat interaktív úton szerkeszteni úgy, hogy a diagrammok információtartalma adatbázisban volt tárolható, és az adatbázis lekérdező rendszerének segítségével analizálható. Két évvel később a Boeing Co. a CYBERNET hálózaton keresztül elérhető hasonló célú rendszert valósított meg (IDEF0 + IDEF1), melyet a 2.1.3 fejezetben röviden értékeltünk. Az azóta létrehozott újabb rendszerek [BIE78], [MUN83] lényeges eltérése a jelen rendszer koncepciójától, hogy csak egyetlen modellezési módszer támogatására épülnek, s nem készülnek föl többfajta tervezési eszköz összeépítésére. Az ISDOS projekt jelenleg folyó továbbfejlesztése tudtunkkal az egyetlen, mely a többfajta modell elvét igyekszik érvényesíteni a specifikációban.

(1) Eugene Merchant a Cincinnati Milacron távl.fejlesztési, a COCAM bizottság tagjának szóbeli közlése 1978, Budapest.



- 4.6 ábra -

4.1.2.3 A rendszerrel szerzett tapasztalatok összefoglalása

Létező módszerek összekapcsolásának segítségével össze lehetett ötvözni egy információs rendszerek tervezése céljára készült, számítógéppel segített módszert egy általános célú, gyártórendszertervezésre a gyakorlatban is alkalmazott modellezési eljárással. Ennek eredményeképpen a gyártórendszertervező módszertan háttéréül jó dokumentációs adatbázist nyerünk. Az ISDOS PSA [BAS75] gazdag analízislehetőségeivel nemcsak a teljes rendszermodell szöveges dokumentációjára ad módot, hanem a rendszerterv kialakítása közben ellenőrzési és útmutató információ is nyerhető segítségével.

Néhány példa:

- A KWIC index, mely az összes előforduló nevet (a hasonló nevű könyvtári ismertetőkkel azonos módon) permutálva jeleníti meg. Ez a funkció használható (pl. két azonosnak szánt, de mégis különbözőre sikerült elnevezés felfedezésére).
- Teljességi analízis egyes elemei:
 - minden adatnév visszakeresése, mely ismert, de nem explicite definiált,

-- azon funkciók (ill. adatok) visszakeresése, melyeknek sem szöveges leírásuk, sem struktúradefini-
ciójuk nem adott.

A rendszer kétségtől hiányossága, hogy a SADT modellek teljes struktúrális analízisét nem lehet az ISDOS rendszerrel automatikusan elvégezni. Ennek oka egyrészt az, hogy a SADT eredeti megfogalmazása hiányos volt (ami azonban a kézi, vagy ember által vezérelt analízist kevésbé hátráltatta), másrészt a már említett ábrázolási nehézségek (nyílhálózat reprezentálta relációk).

4.1.3 A választott módszerek és kifejlesztett eszközök
gyakorlati alkalmasságának bizonyítása, megoldásra
váró problémák felfedezése

A gyakorlati bizonyítási eljárásnak több célja volt, nevezetesen:

a./ A választott modellezési módszer alkalmasságának
kimutatása valódi problémák feldolgozásán keresztül.

b./ A számítógépes tervezési eszközök felhasználhatósá-

gának és korlátainak kimutatása.

c./ Tapasztalatszerzés az eszközök fölhasználásra épülő módszertan kidolgozásához.

Az így nyert ismeretek vezettek a 4.1.4-ben összefoglalt eredmények felismeréséhez. A számítógépi eszközök bevezetése a várakozásnak megfelelő kísérleti eredményeket hozott: az alkalmasság demonstrálása mellett itt is nyilvánvalóan megmutatkozott, hogy a papír-ceruza módszerekkel versenyre kelni kívánó CAD rendszer hozzáférése döntő kérdés. A tervezőmunka folyamatában és nem végén (vagy szakaszainak legvégén) kell az eszközöket a tervező kezébe adnunk. Ez a feltétel a kísérleti bizonyítás során nem volt megoldható, ami az ISDOS rendszer több hasznos szolgáltatásának kihasználását nem tette lehetővé, t.i. azokat, melyek a tervezési munka során véletlenszerűen jelentkező információigényt hivatottak kielégíteni.

4.1.3.1 A funkcionális modellezési eljárás alkalmazása számítógépi programrendszer tervezésére [BER79a]

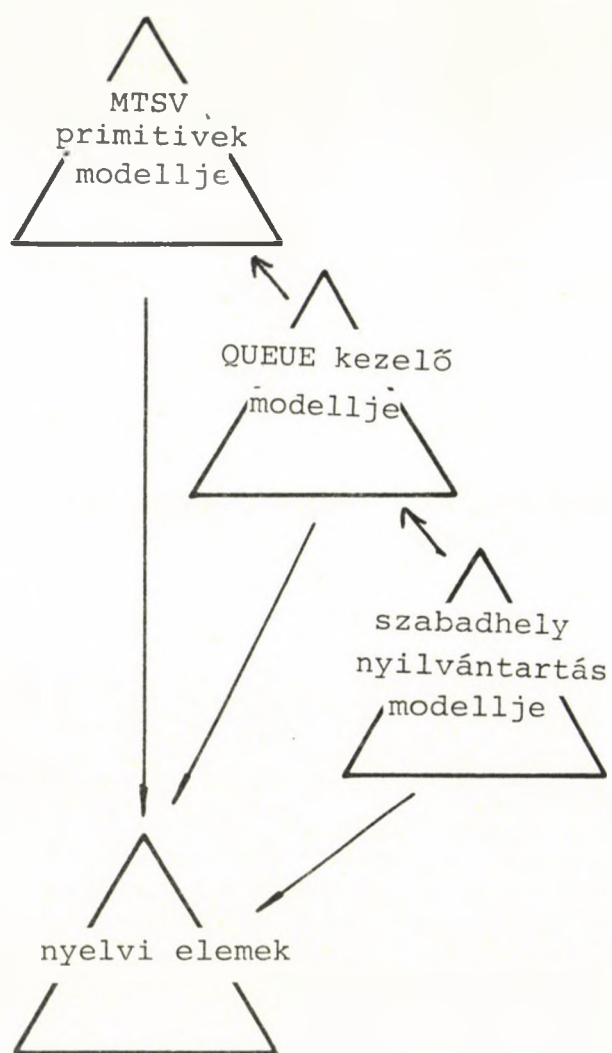
Cél: A választott funkcionális modellezési eljárás első kipróbálása valós méretű feladaton.

Feladat: Egy gyártórendszer anyagmozgató funkcióit vezérlő berendezésben futó multitask operációs rendszer (MTSV) magjának megtervezése és megvalósítása.

Eredmények: A feladat elegendően bonyolult volt ahhoz, hogy a funkcionális tervezési módszernek megfelelően leírása csak több modell segítségével történhetett 4.7. ábra. Az egyes modellek közti mechanizmuskapcsolat tisztán a 4.1.2.2 szerinti értelmezésnek felelt meg, a mechanizmus másfajta értelmezése ennél a feladatméretnél nem volt szükséges. (T.i. a funkcionális és realizációs struktúra nem volt szükségszerűen elkülönítendő.)

A szokásosnak tekinthető rétegződés (melyet a három modell ábrázol) megkönnyítette a legfelső modell alkalmazásspecifikus funkcióinak olyan lebontását, ahol a legalsó szintű funkcióknak legvégül vagy a megvalósító programozási nyelven, vagy a mechanizmusmodellben egyszerűen meg lehetett találni a megfelelőjét. A 4.1.4 pontban látni fogjuk, hogy nagyobb rendszer-méret és szélesebb szakterületi háttér esetében ez a megfeleltetés elvi gondot jelent.

Fölvetett kérdések: Lehet-e teljesen, vagy részben automatikus realizációs stratégiát vagy stratégiákat kidolgozni, melyet



- 4.7 ábra -

tisztán funkcionális modellekre alkalmazhatunk?

Mi a fogalmi viszony a realizáció és a teljes funkcionális felépítmény (illetve modellek) között?

4.1.3.2 Kisméretű ember-gép rendszer modellezése [PAL78]

Cél a modellezés véghezvitele az adatbázis és analízis kapcsolat felhasználásával.

Feladat: Egy konveyor soron függesztett lemezek mozognak. A festésüket végző, munkásból és robotból álló rendszer modellezését kellett elvégezni, a robottal szemben támasztandó funkcionális követelmények kimutatására.

Eredmény: A funkcionális analízist egyetlen modell földállítással hajtottuk végre. Az adatbázisba vitel a modell struktúrájának explicit ábrázolási képességén keresztül (PSA Formatted Problem Statement) teljes keresztreferenciát adott arról, hogy

- mely funkciók végrehajtásához szükséges a robot ill. a munkás;

- melyek a már ismert, de még definiálatlan struktúrájú dolgok (a lebontás folyamatának irányításához folyamatosan használt információ);
- melyek a teljes rendszer megvalósításához szükséges mechanizmusok;
- melyek azok a funkciók, melyeknek megvalósítási lehetőségéről még nem intézkedtünk.

A feladat szempontjából természetesen elsőrendű előny volt a funkcionális modellezés alaptulajdonsága, nevezetesen a funkciók megfogalmazása a végrehajtó eszközök előzetes megkötése nélkül. Ez több változat kidolgozását tette lehetővé (funkciók különféle megosztása az ember és a robot között). Az automatizált rendszer nyilvánvaló előnyeit, a naprakészen tartott szöveges és grafikus dokumentációt is az eredmények közé kell sorolni, bár ennek elsősorban nagyobb analízisfeladatok esetén van igazán nagy jelentősége.

Főlvetett kérdések: A modellezési módszer imént említett előnye (a realizációfüggetlen funkciótervezés) ebben a feladatban jól kihasználható volt. Visszavetítve azonban a kérdést a 4.1.3.1. feladatra, kérdésessé tette, vajon -- ellentétben a

jelen feladattal -- miért volt gyakorlatilag szükség ott a mechanizmus-primitívek ismeretére a funkcionális tervezés elvégzéséhez. A két modellezési feladat világította meg a mechanizmus kettős természetét és adta az alapot a 4.1.4-beli két definíciónak: a mechanizmus és a funkcionális tervezés meghatározásának.

A modellezett rendszer mérete miatt -- az analízishez megkövetelt mélységet figyelembe véve -- nem volt föltétlenül szükséges a funkcionális modellek közötti ellentmondások kiszűrésére automatizmusokat használni, azonban néhány olyan tényre derült fény a modell kézi analízise folyamán, mely egy teljes és automatikus analízis megtervezéséhez használható föl. Így:

- Nyílfolytonosság ellenőrzése az egyes diagrammok között. Az egyes diagrammok ugyanis redundanciát -- és így potenciális ellentmondást hordoznak e tekintetben.
- A nyílhálózat akkor is mond valamit az adatok struktúrájáról, ha datagrammok nem is készülnek. Ebben a tekintetben ellentmondás lehet mind a datagrammok és aktigrammok között, mind csupán aktigrammokon belül.

- Kimutatható volt példán keresztül, hogy egyes esetekben elvileg sem dönthető el, hogy a nyílhálózat által ábrázolt relációk teljességi hiányra, vagy ellentmondásra utalnak. Ez fontos következtetés egy lehetséges analízisprogram tervezése szempontjából.

4.1.3.3 A módszer kísérleti felhasználása nagy rendszer analízisére

Cél: Valódi, nagyméretű rendszeren ki kellett próbálni a modellezési módszert, tapasztalatot kellett szerezni a követendő modellkészítési módszertan kidolgozásához. Emellett a gyakorlatban is bizonyítandó volt a SADT grafikus rendszer alkalmassága nagyméretű feladat kezelésére.

Feladat: A Csepel Szerszámgépgyár IGYR 630 integrált gyártórendszerének utólagos elemzése [BER80]. Az elemzés a fenti célok mellett egy átfogó rendszerterv születésére és a konkrét rendszer még kialakítatlan és már meglévő funkciói közötti ellentmondások feltárására tört.

Eredmények: A modellezési feladat annyiban eltért a valós alkalmazási körülményektől, hogy retrospektíve végeztük el: egy már megvalósítás közben lévő rendszeren. Ez a funkcionális

modell felállításánál nemcsak könnyítést jelentett, hanem nehézséget is okozott azáltal, hogy igen sok megkötést kellett a redlis leírás készítéséhez figyelembe venni.

Ez a feladat vezetett a rendszerek funkcionális rétegződésének a következőekben megfogalmazandó elvéhez, továbbá a fentről lefelé és alulról fölfelé végzett tervezési lépések közötti összefüggések és az innovációs folyamat közötti kapcsolat feltárásához (4.1.4 fejezet). A rendszerterv grafikus dokumentációja a számítógéppel segített rendszeren készült.

Fölvetett kérdések:

A tervezési folyamat iránya ennél a feladatnál hol felülről lefelé, hol alulról lefelé haladt. Mivel ezt nem tartottuk helyesnek, (1) közbenső megoldást kellett keresni. Ezt a megoldást a 4.1.4.1 fejezet tárgyalja.

A tervezési folyamat során többször is fölvetődött a kérdés, mikor kell, vagy lehet a lebontás folyamatában megállni anélkül, hogy a hiányos tervezés vagy a túlspecifikálás hibájába esnénk. A kérdés megválaszolásához a funkcionális

(1) Nem számítjuk alulról fölfelé tervezésnek azt az esetet, amikor a redlikből absztrakció útján nyert fogalmakat használunk a tervezésnél.

tervezésnek az eddig ismertnél pontosabb definícióját kellett kidolgozni (ld. 4.1.4.2 fejezet).

A méretből adódóan sok modellel lehetett csak a tárgyszerkezetet leírni. Az már az eredeti SADT publikációkból is ismeretes (és intuitív is világos) volt, hogy többféle nézőpontból kell ugyanazt a szerkezetet modellezni ahhoz, hogy a megkívánt funkciókat ténylegesen specifikáljuk. A készítenő modellek nézőpontjaira és a modellek rendezésére alakítottuk ki a 4.1.4.3 elvet, mely a szerkek rétegszerkezetét és többnézőpontú modellezését egységes alapon kezeli.

Ennél a feladatnál már nem volt igaz (legalábbis nem előre láthatóan) a fizikai és funkcionális felépítmény azonossága, így rákényszerültünk a SADT mechanizmuskapcsolatának részletesebb vizsgálatára és a kétféle mechanizmusreláció következetes definiálására (ld. 4.1.4.4 alfejezet).

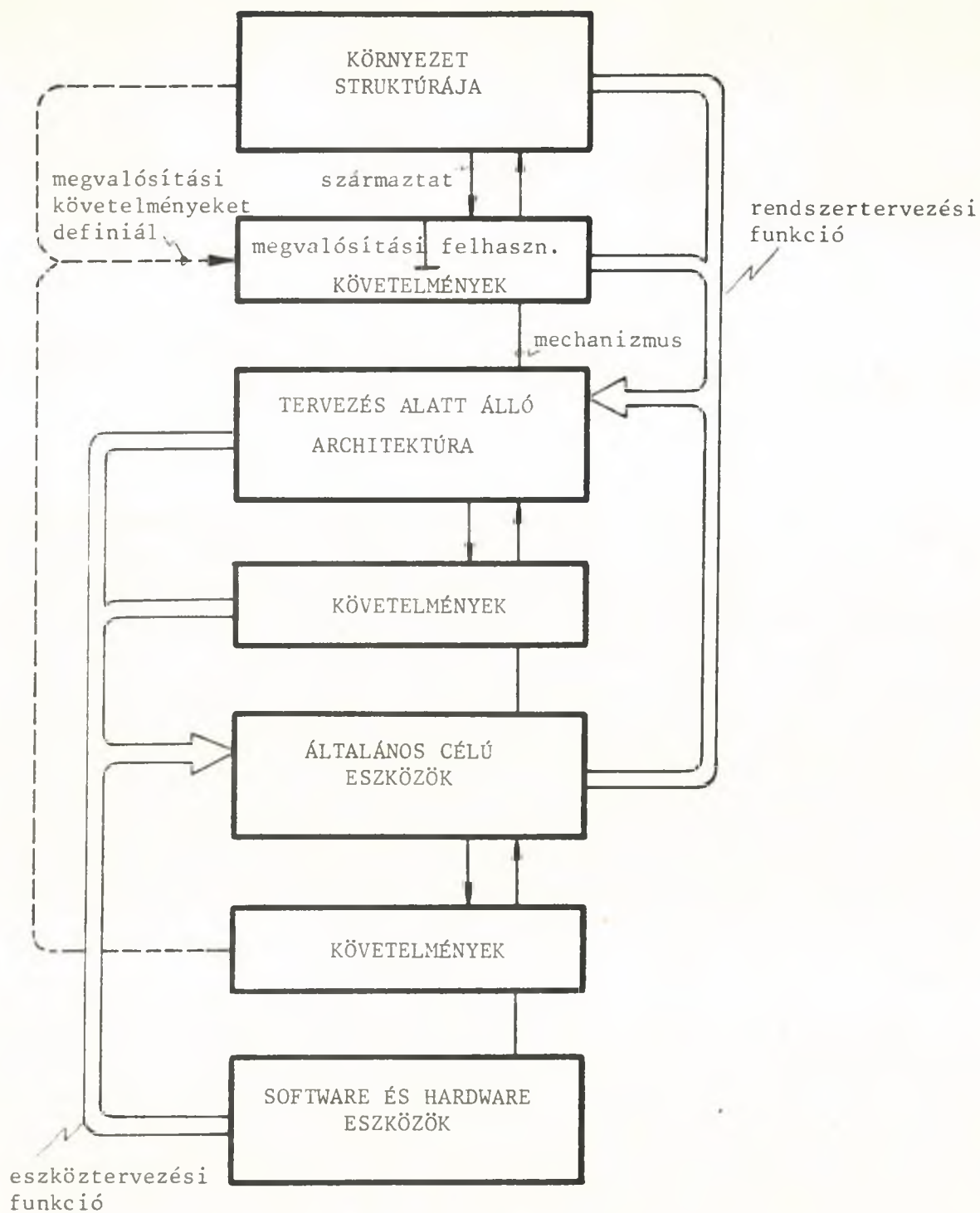
A modellezési gyakorlat bonyolultabb szerkek esetében a SADT áttekintést növelő hatása ellenére felveti a szerkezetmodellek áttekinthetőségi határainak kérdését. Ennek megválaszolására tesz kísérletet saját eredmények alapján a 4.1.4.5 alfejezet.

4.1.4 A funkcionális modellek továbbfejlesztésének néhány eredménye

4.1.4.1 A tervezési folyamat iránya, minőségi követelmények az eszközökkel szemben

A tervezési folyamatban alulról felfelé és fentről lefelé végzett lépések váltakozva követik egymást. Az a sokat tárgyalt ellentmondás, miszerint a fentről lefele tervezett rendszereknél semmi sem biztosítja, hogy az eredmény megvalósítható legyen, az alulról felfelé végzett tervezés pedig nem eredményez a kívánalmaknak megfelelő funkciókat, föloldható az alábbi módon (4.8. ábra). A következtetés célrendszerek tervezésére érvényes, tehát arra az esetre, amikor egy konkrét rendszert kell megtervezni adott környezeti feltételek és eszközbeli kötöttségek figyelembevételével.

Egy másfajta, szintén funkcionális analízist igénylő feladat lehetne pl. több rendszer analízise abból a célból, hogy meghatározzuk az eszköztár fejlesztésének valamely cél szerint optimális stratégiáját [BER81a]. Ugyancsak érdekes feladat lehet egy eszköztár analízise annak megállapításra, hogy az mennyiben teljes, illetve, hogy meghatározzuk a belőle előállítható rendszerek egy osztályát.



- 4.8 ábra -

A tervezés folyamatában a követelmények meghatározásakor ismernünk kell a lehetséges eszközök (diszciplínák, az adott szakterület mindenkori fejlettségi foka, "state of the art") által szabott korlátokat és lehetőségeket. Ezen megkötések széles tartománya képzelhető el: egy-egy konkrét megvalósítási részlet előírásától (pl. pótlólagos automatizálás) egészen a tervezhető rendszertulajdonságok globális lehetőségeinek felsorolásáig (pl. egy kiemelt erőforrás egyediségére tett kikötés) [BER81b]. A megkötések száma tendenciájában annál nagyobb, minél egyedibb hatáskörű megszorításokból tevődik össze. A kötöttségeknek két olyan tulajdonsága van, melynek megléte elképzelhetővé teszi, hogy a funkcionális tervezés folyamán ne kényszerüljünk egyedi döntésekre, hanem bizonyos általános irányelvekre hagyatkozhatunk, amikor a funkcióknak részletfunkciókra bontásakor a megvalósíthatóság irányába akarunk lépni. Ez a két tulajdonság a megkötöttségek általánossága és függetlensége.

Az általánosság követelménye szerint, ha egy funkció létezik és van értelme használni (szemantikusan nem hibás a kontextus) akkor lehessen is használni -- legfeljebb más mennyiségi paraméterek mellett. Ebben az értelemben az általánosság azt jelenti, hogy a szemantikailag értelmes rendszerállapotok mind elérhetőek legyenek, tehát az elképzelhető és a (legalábbis

elvben) elérhető állapottér egybeessen.

A megkötések funkcionális függetlensége azt jelenti, hogy ha két funkció értelmesen kombinálható, akkor gyakorlatban is kombinálható legyen, bár mennyiségileg nem biztos, hogy azonos eredménnyel (pl. hatásfok).

Az elképzelhető állapotteret kifeszítő független primitíveket geometriai analógiával ortogonális rendszernek nevezzük. Az ortogonalitás megléte azt eredményezi, hogy a környezeti követelményekből kiinduló funkciólebontás egyedi megvalósítási követelményektől mentes lesz, így ténylegesen tisztán lebontó jellegű lehet. A gyakorlatban -- ha ez nem teljesül -- (márpedig nem teljesül) elő kell állítani azt a lehetőleg minimális számú megvalósítási követelményt, melyet a lebontó jellegű tervezésnek figyelembe kell vennie. Ezeknek mint peremfeltételeknek ismeretében majdnem tisztán felülről lefelé bontás végezhető. Többdiszciplináris rendszereknél, közöttük a gyártórendszerek esetében is ezek a peremfeltételek csak a szakemberek gyakorlatában testesülnek meg, explicit kifejezést esetleg nem is nyervén.

A funkcionális tervezés szemantikai folyamat, mely a bonyolultabb fogalmat részeire bontja. A lebontásnál csak akkor van lehetőségünk a részek megnevezésére, ha ismerünk egyszerűbb szemantikájú alapvető fogalmakat stb., mígnem elérünk a primitív fogalmakig. Éppen ezek azok a fogalmak, melyeknek ortogonalizálását célul tűzzük ki. A fizikai eszközök ortogonalitását is lehet definiálni, de nem önmagukon belül, hanem felhasználásuk (a funkciókkal fennálló viszony) szemszögéből: akkor rendelkezik az eszközkészlet evvel a tulajdonsággal, ha a primitív funkciók tetszőleges szemantikailag helyes kombinációja realizálható segítségükkel. (Ez a definíció már nem folytatja teljes mértékben a geometriai analógiát.)

Az eddigiekből az is következik, hogy a funkcionális felépítmény (a funkcionális tervezés végeredménye) mindig csak viszonylagosan teljes: ismerni kell a teljességi kritérium megadásához a felépítményt befogadó környezet fogalmkészletét. Mindezek alapján az alábbi módon definiáljuk a funkcionális tervezést:

A funkcionális tervezés lényege, hogy a rendszer feladatát visszavezeti olyan (absztrakt) alapfunkciókra és dolgokra, melyeknek valamely diszciplínán belül kialakult szemantikájuk van. Kialakult szemantikán itt azt kell érteni, hogy egy

Összetett funkció neve azonos egy fizikai jelenségek által definiált funkciók rendszerének nyelvi jelével, továbbá a diszciplína ismer olyan fizikai rendszert (vagy létrehozásának módját), mely - a diszciplínán belül - bizonyíthatóan a funkciók eme rendszerét hordozza.

A definíció önmagában hordozza a teljesség kritériumát is. Figyelemreméltó, hogy ezek szerint diszciplínáris tudás nélküli funkcionális tervezés logikai lehetetlenség (diszciplínáris tudáson érte a szakterületek absztrakt fogalmainak értését), hiszen csak végső soron ismert fogalmakra visszavezetett funkcionális felépítmény definiálhatja a tervezett rendszert. Természetes, hogy a diszciplínák nagyon elvont funkciófogalmakat is kialakíthatnak (esetleg multidiszciplínárisat), ezért lehetséges, hogy az egyes tervező, mint egyén a diszciplínák részletes ismerete nélkül is teljes rendszerdefiníciót adhat. A funkciófogalmak mögött azonban mindig ott van egy vagy több szaktudomány definíciója, vagy tárgyi referenciája.

Az ebben a fejezetben ismertetett elvet alkalmaztuk a formalizált jelölésrendszerre, ami elvezetett a szemantikai analízis 4.2-beli megfogalmazásához.

A szemantikai definíciót legalsó szinten tárgyi referenciákra

vezettük vissza, ezért azt értelmi definíciónak kell felfogni. Ugyanekkor a fogalmak kombinációja jelentéssel bíró, de tárgyi referenciával már nem bizonyosan rendelkező fogalmakat hozhat létre, így a funkcionális felépítmény jelentéssel rendelkező specifikációkat állít elő. Nem foglalkozunk itt a realizálhatóság elvi feltételével, csupán megjegyezzük, hogy a funkcionális tervezés módszertana a gyakorlatban úgy hidalja át ezt a nehézséget, hogy a tervezést megvalósítási gyakorlattal is rendelkező tervezőcsoportra bízza, akik a specifikációt és tervezést iterálva jutnak el a végső megoldáshoz. Nem lehetetlen, hogy a realizálhatóság vizsgálata a jövőben ilyenirányú (beépített vagy folyamatosan szerzett) szakismerettel felruházott automatikus rendszerekre lesz bízható.

4.1.4.3 A rendszerek rétegszemlélete

Nagy rendszerek tipikus felépítése egymásra épülő szintekből áll, melyben minden szint csak az alatta lévő szintre épül rá (a szintek takarják egymást) [PAR72], [PAR78]. A gyártórendszerek modellezése során kialakult rétegszemlélet ennek az elvnek az érvényesítése a funkcionális modellezés területén. Komplex gépipari rendszerek egyedeinek tipikus rétegződését az alábbi módon lehet leírni. A leírás annak feltételezéséből indul ki, hogy a modellezendő funkciók körébe elvben a

rendszer élete folyamán előforduló minden funkció beletartozhat.

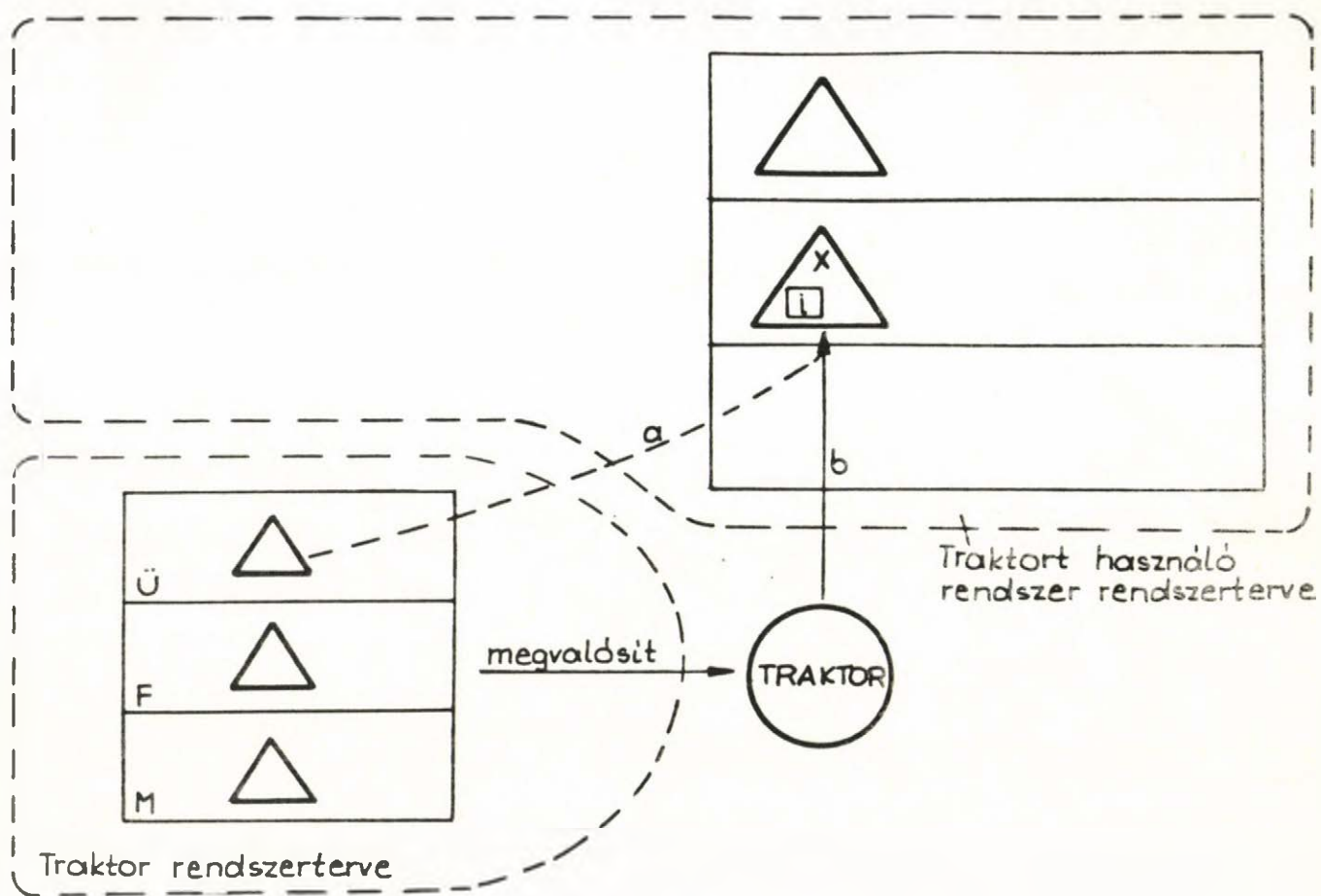
A legfőbb rendszerreteg az egyedi rendszer életfunkcióit írja le - szélsőséges esetben az ötlet felmerülésétől a megsemmisülésig. (A gyakorlatban esetleg csak a rendszer megvalósítása utáni funkciókra szorítkozunk.) A legfelső rendszerreteg tehát a fejlődési-, vagy a megszorítást elfogadva üzemelési réteg. A rétegen belül a funkciók lebontásának nézőpontja: a rendszer fő állapotait és ezek átmeneteit kell egymástól elkülöníteni. (Így a rendszergenerálás, üzemeltetés, karbantartás, ki- és bekapcsolás, üzemmódok közötti átd állás stb.) Az üzemelés szintjének leírásához -- annak bonyolultságától függően -- több modell is szükséges lehet, a nézőpont azonban azonos.

A következő szint a funkcionális réteg. Ebben helyezkednek el azok a modellek, melyek a rendszer fő üzemállapotaiban végzendő (a rendszer céljának megfelelő) funkciókat tárgyalják. Ez egyben a funkciók bontását korlátozó nézőpont. (Természetesen elképzelhető, hogy egy-egy itt szereplő modell a felsőbb szint több funkcióját is képes ellátni). Ez a réteg igen sok egymásraépülő modellt is tartalmazhat, de, általános célú eszközfunkciókat nem tárgyal, azokra csak mint egy alsóbb réteg eszközeire hivatkozik.

A mechanizmusok, vagy eszközök rétege a harmadik, legalsó szint, mely azokat az általános célú eszközfunkciókat tárgyalja, melyeknek felhasználására a tervezett rendszer épül. Nézőpontja ennek megfelelően arra irányul, hogyan lehet az eszközöktől elvárt funkciókat lehetőleg általános és egymástól független módon részekre bontani. A legalsó szint a szakterületen jól definiált fogalmakig hivatott elérni a lebontással. Ez a gyakorlatban igen eltérő mélységű lehet a rendszer egyes pontjain. A 4.9. ábra mutatja, hogy az egyes eszközök modelljei önmaguk is független rendszerek modelljeként képzelhetők el, több belső réteggel. A hármas tagozódáson túl tehát további résztagozódások is bevezethetők. Lényeges, hogy a mechanizmusréteg funkcióit különválasszuk a funkcionális rétegtől, mivel az utóbbiban az ortogonalitás (az egyediség miatt) nem olyan súlyosan latba eső követelmény.

4.1.4.4 A mechanizmuskapcsolat

A SADT mechanizmuskapcsolatát az alábbi módon értelmezzük ([BER81d] IV. fejezet). Két módon is hivatkozhatunk egy funkciót végrehajtó (megvalósító) mechanizmusra. Az egyik mód szerint megadjuk egy olyan funkcionális modell nevét, mely a mechanizmusfunkciót részleteiben taglalja. Másik mód, ha egy olyan fizikai rendszerre hivatkozunk neve segítségével, mely a



- 4.9 ábra -

funkciót képes végrehajtani.

Az első esetben a modellezett funkció és mechanizmusa közötti megfelelés (1) formálisan vizsgálható. A másik esetben csak a mechanizmus létezésének ténye ismeretes, de a megfelelés helyességére nincs bizonyíték. A két esetben a kapcsolat mindenképpen a funkció és mechanizmusa között áll fenn, de a rendszerterv helyességének bizonyítéka a két esetben más és más. A két mechanizmusféleség között a megvalósítás (illetve a jelölésre használt nyelv adott szavának denotációs viszonya) teremt kapcsolatot. A mechanizmus mindig valamilyen dolgot ábrázol, ennek megnevezése történhet akár mechanizmusmodellbeli funkciójának, akár megvalósításának a neve alapján. Az is előfordulhat a gyakorlatban, hogy e két név azonos (különbségtételre a kontextus alapján van mód).

A megvalósítás mindig valamilyen fizikai struktúra létrehozásával jár, mely műszaki kategóriákban ölt testet. Így róla esetenként, vagy területenként kell megmutatni, hogy milyen fizikai dolgok (minőségek, mennyiségek) között milyen funkcionális kapcsolatot hordoz. Egyes területeken kialakulhat a

(1) A formális egyezés feltétele, hogy a hívó funkció minden be- és kimenő nyílának feleljen meg a hívott funkció egy be- ill. kimenő nyila. Ezen megfeleltetés implicit módon a hívó modell dolgaihoz is mechanizmust rendel, így annak adatlebon-
tásával sem állhat ellentmondásban. A mechanizmusmodellnek ezenkívül természetesen lehetnek egyéb be- és kimenetei is.

4.1.4.5 Az áttekinthetőség határai ([BER81d] IV. fejezet)

A funkcionális modellezés elsőrendű célja az volt, hogy segítségével az ember az addig áttekinthetetlenül nagy rendszereket kezelhetővé tegye. Az eddigiek csak egy meghatározott nagyságrenddel tolták előre a kezelhetőség határt, azonban nem adtak felvilágosítást arra nézve, hogyan lehet ezt a határt tetszőleges mértékben tágitani. Most arra keresünk választ, hogyan lehetne a modellek összefüggésének bonyolultságát magával az eredeti módszerrel megragadni, hiszen így rekurzív módszer birtokába jutnánk, mely, ha újból akadályba ütközik, önmagára alkalmazva győzne le azt.

A modellstruktúra ábrázolásának módosíthatása önmagában nem is ígér ilyen lehetőséget. Vegyük figyelembe, hogy a funkcionális struktúra az ábrázolt rendszer teljes életciklusának csak egy körülhatárolt szakaszt ragadja meg. Ha mármost a teljes életciklus összes funkcióját modellezzük, akkor azt tapasztaljuk, hogy az eredeti modellstruktúra megvalósítási funkciók és közöttük meglévő dologi kapcsolatok formájában egy újabb, a rendszert magát leíró funkcionális felépítménynél magasabb szinten jelenik meg. Az eredeti modellkapcsolatok be- és kimeneti kapcsolatokon keresztül struktúrázhatóak és szükség esetén több modellrétegben ábrázolhatóak. Ha pedig az imént

funkciók és kapcsolódó dolgok megvalósításának tárháza, műszaki kategóriákkal leírva mind a funkciók, mind a dolgok mechanizmusát. Ezt a műszaki tervezés már a meggondolások újraismétlése nélkül fölhasználhatja bonyolultabb fizikai felépítmény létrehozásához. A fizikai struktúra ezek szerint egy bizonyos szint fölött egyezni fog a funkcionálissal. E szint alatt pedig a megfeleltetés esetleg csak esetenkénti, szakterületi bizonyítással érhető el: a fizikai és funkcionális struktúra elválik egymástól. A mechanizmusnevek tehát a fizikai valóság elemeinek referenciái, így mind a funkciók mind a dolgok esetén dologi természetűek.

Fel kell hívni a figyelmet arra, hogy a fizikai rendszer nem egyszerűen egy mechanizmus, hanem általában mechanizmusok egy rendszere, amely csak az adott összefüggésben (a hozzárendelések adott kiosztásában) helytálló.

A gyakorlati tapasztalatok alapján az irodalomból ismert elveket néhány új elemmel kibővítettük, és kidolgoztuk azt az eljárást, amelynek segítségével a funkcionális tervezési módszertan a gyártórendszerek életfolyamatának e kezdeti szakaszába beilleszthető. A módszertant kézikönyvszerűen rögzítettük ([BER81c], [BER81d]) és ipari környezetbe is átadtuk. Első ipari alkalmazása egy felügyelet nélküli gyártócella funkcionális tervezése volt ([BER81e], [BER82d]).

kapott modellek közötti kapcsolatok újra túlmennek az áttekinthetőség határára, az algoritmus önmagára alkalmazható.

Ennek az elvnek egy másik megfogalmazása szerint egy bizonyos összetettség felett csak úgy lehet a megértésben előrelépni, ha nemcsak az elemzett rendszer funkcióit értjük meg (és modellezzük), hanem fejlesztésének, megvalósításának, használatának és kiértékelésének funkcióit is. Ezt a megfogalmazást pedig a műszaki fejlődés történetének tapasztalataiból leszűrt bizonyítékokkal lehetne alátámasztani. A tervezési-gyártási folyamat ilyen jellegű leírását adta [EAT77].

Jelen fejezet saját eredmények alapján azzal foglalkozik, hogyan lehet a funkcionális leírások formális definícióját megadni (4.2.1), majd értékeli a funkcionális leírások által létrehozható szemantikai tartalmakat (4.2.2), végül bemutatja az elmélet által lehetővé tett funkcionális analízist (4.2.3) és szól arról, hogyan lehet a tárolást és analízist egy relációs adatbázis segítségével elvégezni (4.2.4).

4.2 A funkcionális notáció és adatbázis kapcsolata

Mindaddig, míg a funkcionális modellek helyességének formális vizsgálatával nem foglalkoztunk, nem volt szükség a funkcionális architektúra belső törvényeinek alaposabb vizsgálatára. Ezt azért tudtuk elkerülni, mert a funkcionális leírások grafikus ábrázolása az esetek zömében a mérnök olvasó számára egyértelmű kifejezési eszköz. A leírások elsődleges információtartalmának adatbázisba vétele, visszakeresése nem tette elengedhetetlenné a készített rendszerleírások tartalmi ellenőrzését.

A gyakorlatban elkészített modellek így a készítő és olvasók közötti megegyezést (illetve többféle szakmai ismerettel rendelkező csoport belső megegyezését) szolgálták. A nagyméretű tervekkel kapcsolatban csak ezután merült föl a kérdés: milyen

alapon bizonyítható, hogy a terv teljes és hibátlan. A probléma méretére gyártórendszerek esetében jellemző, hogy több száz vagy ezer funkció és dolog (anyag, információ) definícióját, kapcsolatának leírását kell megvizsgálni. Ennek megfogalmazásához már formális eszközökre van szükség.

Az itt kifejtett notáció alapja egyrészt a 4.1.4 fejezetben intuitíve megmutatott modellszemlélet, másrészt az a cél, hogy relációalgebrai eszközökkel végezzük a formalizálást (t.i. ekkor az analízisek elvégzéséhez a relációs adatbázisok eszköztárára támaszkodhatunk).

4.2.1 A funkcionális notáció formalizálása

Definíciók és alapfogalmak

Ebben a fejezetben bevezetjük a funkcionális architektúra (FA) alapfogalmait és az ezeken értelmezett relációkat. Ez utóbbiak segítségével új jelentések (szemantikus tartalmak, közlések) generálhatóak.

Az alapfogalmak bevezetésénél a valós világra mutató referenciát igyekeztünk adni azért, hogy a modellezési módszer valós rendszerekre alkalmazásánál ezek szolgáljanak támpontul. Ha-

sonlóan tesszük ezt, mint ahogy a geometria "egyenes" fogalmát a fénysugár útjához, vagy a kifeszített fonálhoz mint referenciához szokták magyarázatképpen kötni -- annak ellenére, hogy a tisztán formális definíció céljából erre semmi szükség nincs.

4.2.1.1 Funkcionális architektúra (FA)

A funkcionális architektúra információközlésre szolgál. A szemantikai definíció az ismeretlen szemantikájú egészet ismert szemantikájú alapegységekre vezeti vissza. Ismertnek tételezzük föl a szemantikát, ha az információ adó és vevő eleve közös referenciamodellel rendelkezik, melynek denotációja (pl. egy funkció neve, vagy elfogadott szöveges magyarázat) elegendő az információ átadásához.

Nem tévesztendő össze egy rendszer referenciamodelle a rendszer valamely realizációjával. A referenciamodell esetleg csak elvben létezik (pl. szimulációs modell formájában), de nem biztos, hogy fizikailag is jelen van. Bár egy megvalósítás adott esetben lehet referenciamodell, két különböző fizikai rendszer funkcióinak kombinációja nem feltétlenül jelenti a két rendszer fizikai kombinációját. Ehhez esetleg a fizikai rendszerek valamely modelljének kombinálására van szükség. Egy realizáció csak akkor lehet referenciamodell, ha biztosítva van identifikációjának lehetősége, tehát a rá, vagy részeire hivatkozás azonossági relációkon keresztül teszi lehetővé az információ közlését.

4.2.1.2 Funkció (F_1, F_2, \dots)

Mindenfajta létező, mely csak időintervallumban létezhet, illetve ezek osztályai. Időben működő transzformációk egy osztálya, mely dolgok egy osztályát használja föl és dolgok egy osztályát hozza létre. Egy funkció mindig időbeli általánosítás eredménye. Ezért szokás beszélni egy funkció aktiválásairól, melyek során végrehajtott tényleges transzformáció függ a felhasznált dologi osztályok aktuális reprezentánsaitól.

4.2.1.3 Dolog (D_1, D_2, \dots)

Mindenfajta létező, mely időpillanatban is létezik, illetve ezek osztályai. Dolog az anyag, vagy anyagi létezők egy osztálya. Dolog az információ, adat, esemény és az időpillanat maga is.

4.2.1.4 Bemenet, Kimenet (Input/Output) $D_1 = I(F)$, $D_2 = O(F)$, $D_3 = C(F)$

$D_1 = I(F)$: D_1 bemente F -nek

$D_2 = O(f)$: D_2 kimenete F -nek

$D_3 = C(F)$: D_3 vezérlő bemenete ("kontrollja")
 F -nek

Az I/O/C reláció alapfogalom, fennállásának jelzése a fenti.

4.2.1.4.1 Valódi és vezérlő bemenet, valódi kimenet

Szokás különbséget tenni valódi és vezérlő bemenetek között (I,C). Ennek szemantikai alapja, hogy bizonyos dologi osztályoknak, (pl. esemény, időpillanat) és az időintervallumban lejátszódó funkcióknak speciális kapcsolatuk van: a dolog (vezérlő bemenet) jelenléte aktivizálja az adott funkciót. Az egyszerű bemenet ezzel szemben a funkcióosztály aktiválása során átalakul, és résztvesz a kimenet létrehozásában.

Ilyen értelemben használtuk a funkció mint alapfogalom bevezetésénél a "felhasználás" és "létrehozás" szavakat. Az egyszerű és vezérlőbemenetek között éles határvonalat vonni nem lehet, mert definíciójuk nem két egymást kizáró kategória, hanem átfedő. Olyan funkciók esetén azonban, amelyeknek csak egy bemenetük van, ez feltétlenül vezérlő típusú kell, hogy legyen, t.i. minden funkciónak kell stimulus. Ha ez mégis így, akkor létrehozásától fogva örökké aktív funkcióról van szó, mely saját bemenetét állandóan figyeli, mintegy saját maga számára előállítva a stimulust. Szokásos jelölés a vezérlő bemenet jelzésére a $d=C(F)$ helyett a $d=I_c(F)$ is.

4.2.1.4.2 Mechanizmus bemenet (Logikai kapcsolat a realizációval) $m_f = \text{Im}(F)$, $m_d = \text{Im}(D)$

Ha egy FA által specifikált rendszert meg akarunk valósítani, végsősoron egy speciális referenciastruktúrát kell megalkotnunk. A feladat megoldása műszaki tervezési tevékenységekből áll, melyek minden funkcióhoz és dologhoz valamilyen megvalósítást rendelnek. A valóságban ez nem minden esetben felel meg a fizikai rendszer térbeli particionálásának, hanem egyéb megosztás is elképzelhető. Mindenesetre fizikai entitások lesznek az egyes funkcióknak és dolgoknak a megvalósító. Érdekes, hogy már a FA konstruálása idején több oknál fogva is szükség lehet a megvalósítás (megvalósító mechanizmus) és a megvalósított funkció vagy dolog közötti kapcsolat jelölésére:

Bizonyos funkcionális kapcsolatok azon alapszanak, hogy két funkciót azonos fizikai rendszer valósít meg. (Például egy anyagmozgató szolgáltatásra csak akkor számíthatunk, ha ugyanannak a megvalósításnak az adott funkcióját használjuk küldésre, amelynek a vételi oldalát a fogadásra.)

Sokszor egy rendszeren belül kell ábrázolni egy funkciót az őt megvalósító eszközt előállító funkcióval. Ezek szerint kapcsolatukat is be kell tudni mutatni.

A megvalósító mechanizmus reláció bizonyos dolgokat -- fizikai struktúrákat -- speciálisan köt hozzá egy-egy funkcióhoz, vagy dologhoz. Az I/C/O relációkon kívül ezért bevezetjük az $m_f = \text{Im}(F)$ ill. $m_d = \text{Im}(D)$ relációt. Jelentése: m_f az F funkciónak, vagy részének megvalósítása, m_d pedig D dolognak vagy részének megvalósítása. $R(F) = \bigcup \text{Im}(F)_i$ ($i=1, k$) az F funkció megvalósítása, mely fizikailag k részből áll. Az ünió csak akkor

végezhető el, ha a realizációra mint referenciamodellre főnálló azonossági kritériumok teljesülnek. (1) Néhány speciális eset a gyakorlatban is jelentőséggel bír, ezekre nézve ld. 4.2.2.6.1 és 4.2.2.6.2 alfejezeteket.

4.2.1.5 Határfelület (Csatlakozófelület, Interface)

Egy rendszer definiálásának első és legfontosabb lépése az, hogy a rendszert elhatároljuk a környezetétől. Ez a határvonal elválasztja egymástól a rendszeren belüli és rendszeren kívüli dolgokat és funkciókat. Vannak olyan dolgok és funkciók, melyek nem rendelhetők teljes joggal sem a rendszerhez, sem a környezetéhez, mivel a környezetet és a rendszert kapcsolják össze. Azt a funkciót, mely a környezethez tartozó dolgot hoz létre (vagy használ) határfelületi vagy interface funkciónak nevezzük. Azt a dolgot, melyet a környezethez tartozó funkció hoz létre (vagy használ) határfelületi vagy interface típusú dolognak nevezzük. A határfelületi funkciók és dolgok együttesen alkotják a rendszer határfelületét (2) (Megj.: amint minden funkciónak és dolognak, a határfelületnek is van struktúrája). Gyakorlati példa a kétféle határfelületre: egy

(1) Ha $Q = R(F)$, az nem zárja ki, hogy Q még mást is realizáljon F -en kívül.

(2) Az ISDOS rendszer "INTERFACE" fogalma megfelel az itteni határfelület-fogalom realizációjának.

operációs rendszer primitívjei határfunkciók, egy adatátviteli kapcsolatban az átvitt jel határfelületi dolog. Az esetek zömében mindkét interface típus használatára szükség van.

4.2.1.6 Definíciós relációk, aktigramm és datagramm modellek

A funkcionális architektúra végső soron fogalmakat definiál. Ennek során hivatkozik ismert jelentésű fogalmakra. A fogalmak eme egymásraépülését irányított gráf szemléltetheti. A gráf csomópontjai a fogalmakat jelentik, (pl. f_1 , f_2) élei pedig azt a relációt fejezik ki, hogy az egyik funkció (f_2) résztvesz a másik (f_1) definíciójában: ($f_1 \rightarrow f_2$).

A funkciók definíciós gráfja $DefF = \{f_i \rightarrow f_j\}$

A dolgok definíciós gráfja $DefD = \{d_i \rightarrow d_j\}$

Ha a fogalom meghatározásánál elkerüljük a rekurziót, akkor ez a gráf definíció szerint körmentes lesz. Tetszőleges rendű predikátumkalkulus operátorainak mint generátoroknak felhasználásával ez a megkötés (1) nem okoz gondot a gyakorlatban. A körmentes irányított gráf valamely független fákra bontása olyan fogalmi struktúrát eredményez, melyben a fogalmak vagy

(1) A definíciók rekurziómentessége nem feltételezi például a fogalmak segítségével modellezett algoritmusok rekurziómentességét.

fastruktúrát alkotnak, vagy egy másik fa egy elemét használják föl a definícióhoz. Ezzel a gondolatmenettel akár a funktciók, akár a dolgok fogalmi struktúrája fádba rendezhető. (Ezzel kapcsolatban ld. még 4.2.1.9.1-t.)

A funkcionális architektúra definíciós struktúrái fák: a funktciók fája $\{Tfi\}$ és a dolgok fája $\{Tdj\}$. A funktciók ilyen definíciós fája az alapja az aktigramm modellnek, a dolgok egy ilyen fája pedig datagramm modellnek. Az aktigramm és datagramm modellek részeit a 4.2.1.9.1 fejezet sorolja föl részletesen.

4.2.1.6.1 Része reláció $\langle F1, F2 \rangle$ ill. $\langle D1, D2 \rangle$

ha mind $F1$ mind $F2$ része Tfi fának és

van $F1 \rightarrow F2$ ága

akkor $\langle F1, F2 \rangle$

valamint

ha mind $D1$ mind $D2$ része Tdj fának és

van $D1 \rightarrow D2$ ága

akkor $\langle D1, D2 \rangle$

Szavakban: $F2$ része $F1$ -nek, ha azonos fogalomdefiníciós fában vannak és $F2$ közvetlenül vesz részt $F1$ definíciójában.

ha $F1, F2$ része Tfi fának és

létezik $F1 \rightarrow Fx1 \rightarrow Fx2 \rightarrow \dots \rightarrow F2$

akkor $\langle F1 \backslash F2 \rangle$

valamint

ha $D1, D2$ része Tdj fának és

létezik $D1 \rightarrow Dx1 \rightarrow Dx2 \rightarrow \dots \rightarrow D2$

akkor $\langle D1 \backslash D2 \rangle$

Szavakban: egy definíciós fában létezik $F1$ -ből út $F2$ -be, tehát $F2$ nem föltétlenül közvetlenül vesz részt $F1$ definíciójában.

A reláció inverzének jelölése: $\langle F2 // F1 \rangle$ ill. $\langle D2 // D1 \rangle$.

4.2.1.7 (Struktúra) mechanizmus reláció $M(F)=f$, $M(D)=d$

ha F része $Tfi1$ modellnek és

f része $Tfi2$ modellnek és

van $F \rightarrow f$ út,

akkor $M(F)=f$

valamint

ha D része $Tdj1$ modellnek és

d része $Tdj2$ modellnek és

van $D \rightarrow d$ út

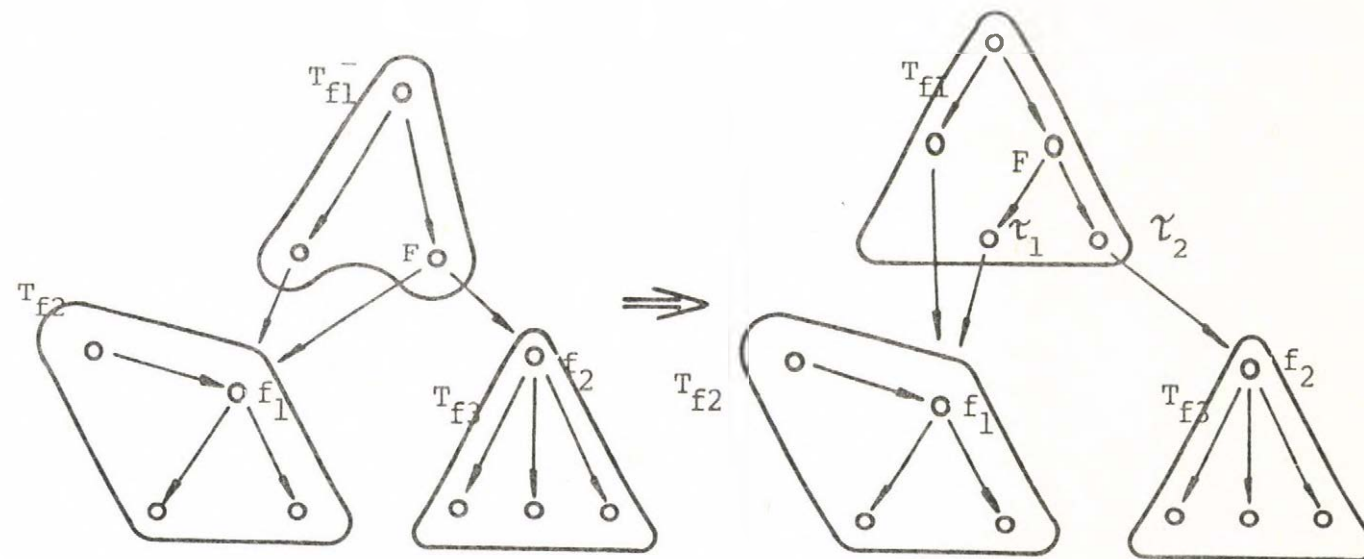
akkor $M(D)=d$.

Szavakban: f egy olyan fogalom, mely F definíciójában részt vesz, de más fogalmakéban is résztvehet. Ezért f egy közös, többszörösen használható mechanizmusfogalom.

a.) Fontos eset, ha $F \rightarrow f$ az egyetlen F -ből kiinduló él T_{fil} -ben, mert ekkor f teljes egészében definiálja F -et. (Hogy ez teljesülhessen, F ill. f egyéb, harmadik féllal fennálló relációi sem lehetnek függetlenek. Erre az M relációk szemantikájának tárgyalásakor még kitérünk.) A "struktúra" mechanizmus azt fejezi ki, hogy f határozza meg F struktúráját (dolgoznál $\bar{a} = M(D)$ D -ét).

b.) Ha ilyen mechanizmusreláció a fogalmi struktúrában nem egyetlen kiinduló él F -nek, ez azt jelenti, hogy F még összetett funkció. Az összetétel és a fák közötti határ átlépése helyett segédfunkciók bevezetésével (ld. 4.10. ábra τ_{a1} és τ_{a2}) elérhető, hogy fák közötti határokat csak az a.) esetnek megfelelően lépjen át a fogalomdefiníciós struktúra. (A példában fennállnak a $f_1 = M(\tau_{a1})$, $f_2 = M(\tau_{a2})$, $F \rightarrow \tau_{a1}$, $F \rightarrow \tau_{a2}$ relációk.)

Ezekután föltételezzük, hogy minden fogalmi struktúrát ilyené alakítottunk. Eszerint a mechanizmusok nem mások, mint a többszörösen használt fogalmi (funkció vagy dologi) struktúrák.



- 4.10 ábra -

Az adatutak funkciók között teremtenek kapcsolatot. Segítségükkel lehet megkötéseket tenni funkciók be- és kimenetei közötti azonossági kapcsolatra. A funkció-funkció közötti anyag vagy információáram kapcsolat fönndálhat két vagy több funkció között. Ha több funkció között áll fönnd, akkor a kapcsolatot egy irányított, több bekezdésű gráffal ábrázolhatjuk, ahol az anyag és információ az élek mentén, azok irányítása szerint terjed. A gráfnak lehetnek belső csomópontjai, melyek nem jelölnek funkciót, mivel rajtuk az információ és anyag változatlan formában terjed tova. A gráf bekezdései a generáló funkciók, levelei pedig a fogyasztó funkciók. Ha a generáló funkciókat megkülönböztetjük a fogyasztó funkcióktól, akkor a gráf körmentes, hiszen minden éle részt kell vegyen valamilyen irányú anyag, vagy információáramban, mely funkciótól funkcióig terjed. A gráf információtartalma ekvivalens az összes lehetséges bekezdéstől-levélig terjedő út információtartalmával. A gráf tetszőleges bekezdéséből induló és leveléhez vezető utat nevezzük adatútnak. Az adatút eleje és vége a funkcióhoz valamely be- vagy kimeneti relációval köthet (I, Ic, Im, O) .

A funkcióutak dolgok között teremtenek kapcsolatot. Segítségükkel lehet megkötéseket tenni a dolgokat felhasználó és létrehozó funkciók közötti azonossági kapcsolatra. Ez a kap-

csolat f  n  lhat k  t vagy t  bb dolog k  z  tt is. Ha t  bb dolog k  z  tt   ll f  nn, akkor a kapcsolatot egy ir  ny  tott, t  bb bekezd  s   gr  ffal   br  zolhatjuk, ahol a transzform  ci   ir  ny  t az   lek mutatj  k. A gr  fnak lehetnek bels   csom  pontjai, melyek nem jel  lnek k  zbens   eredm  nyk  nt kapott dolgot, mivel a be-   s kimen     lek azonos oszt  lyba tartoz   funkci  kat jel  lnek. A gr  f bekezd  sei a transzform  ci  hoz felhaszn  lt dolgok, levelei pedig a l  trehozott dolgok. Ha a felhaszn  lt dolgokat megk  l  nb  ztetj  k a gener  lt dolgokt  l, akkor a gr  f k  rmentes, hiszen minden   le transzform  ci  t jel  l, miszerint a k  r olyan transzform  ci  nak felelne meg, melynek sem bemene- te, sem kimenete nincs. A gr  f inform  ci  tartalma ekvivalens az   sszes lehets  ges bekezd  st  l-lev  lig terjed     t inform  ci  - tartalm  val. A gr  f tetsz  leges bekezd  s  b  l indul     s level  - hez vezet   utat nevezz  k funkci   tnak. A funkci   t eleje   s v  ge a dologhoz valamely be- vagy kimeneti rel  ci  val k  t  dhet (I, I_c, O) . Az I_m rel  ci   itt nem szerepelhet, hiszen dolog realiz  ci  ja is dolog.

Realiz  ci  s funkci     s realiz  lt dolog k  z  tt az adat  thoz formailag hasonl   adat  t   p  thet   f  l, de az adat  t v  g  n funkci   helyett a realiz  lt dolog   ll, m  gpedig az adat  t v  gs   elem  vel I_m rel  ci  ban. Az ilyen t  pus   adatutakra a t  bbt  l elt  r   teljess  gi krit  riumok kell, hogy vonatkozzanak majd,

mert nem feladata a rendszer specifikálójának, hogy a realizáció minden funkcióját és fizikai struktúráját is megtervezze. Mindazonáltal ennek a relációnak is ábrázolhatónak kell lennie a FA-ban.

4.2.1.8.1 Zárt adatutak és funkciótutak (ld. 4.11. ábra)

ha F_1, F_2 része T_{fi} modellnek és

x_1, \dots, x_n része T_{dj} modellnek

akkor $F_1.x_1.x_2. \dots x_n.F_2$ funkciótut

és $x_1 = O(F_1)$, $x_n = I(F_2)$

ha D_1, D_2 része T_{fk} modellnek és

y_1, \dots, y_m része T_{fl} modellnek

akkor $D_1.y_1.y_2. \dots y_m.D_2$ adatút

és $D_1 = I(y_1)$, $D_2 = O(y_m)$

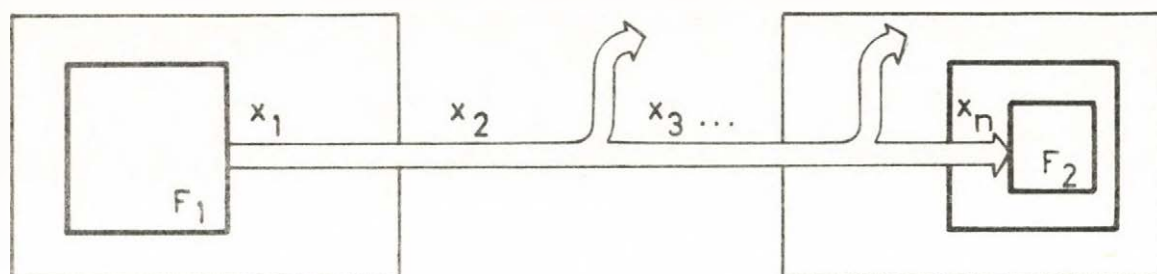
teljes adatkapcsolat $FA.x.FB$

$x = O(FA)$, $x = I(FB)$

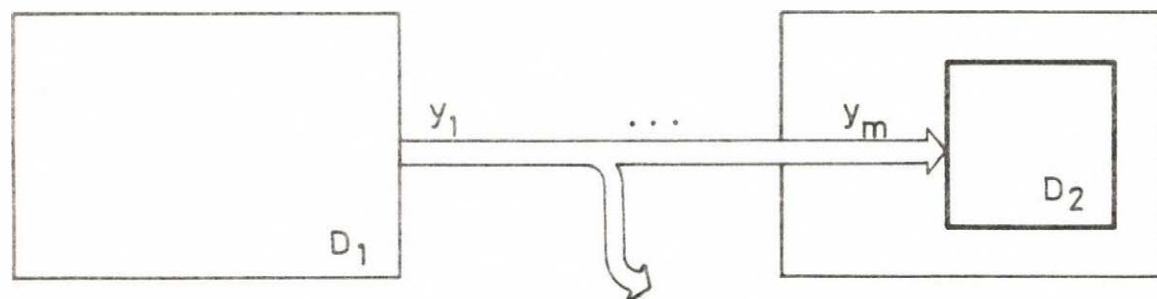
egy funkció létrehoz valamit, amit a másik használ.

részleges adatkapcsolat $FA.x.y.FB$

feltéve, hogy x és y azonos, vagy $\langle x \setminus y \rangle$ vagy $\langle y \setminus x \rangle$



$$F_1 \cdot x_1 \cdot \dots \cdot x_n \cdot F_2$$



$$D_1 \cdot y_1 \cdot \dots \cdot y_m \cdot D_2$$

$\Rightarrow x=O(FA), y=I(FB)$, és x nem független y -től.

Szavakban: egy funkció létrehoz valamit, ami nem független egy másik funkció bemenetétől.

valódi/nem valódi adatút(funkcióút)

csak az az adatút a valódi, melyre

$FA.x_1.x_2. \dots x_n.FB$ -ben fõnndáll, hogy:

minden i, j -re $x_i \& x_j \neq 0$,

egyébként nincs kapcsolat a funkciók között.

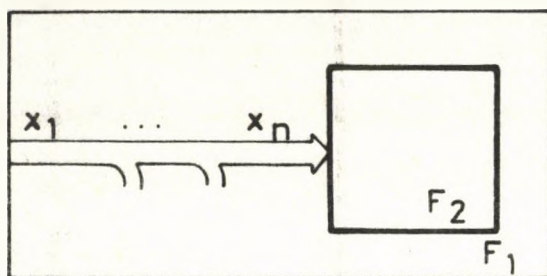
4.2.1.8.2 Nyílt adatutak és funkcióutak (ld.4.12. ábra)

ha $F_1 \setminus F_2$ és
 $-F_1.x_1.x_2. \dots x_n.F_2$ adatut,
 akkor $x_1 = I(F_1), x_n = I(F_2)$;

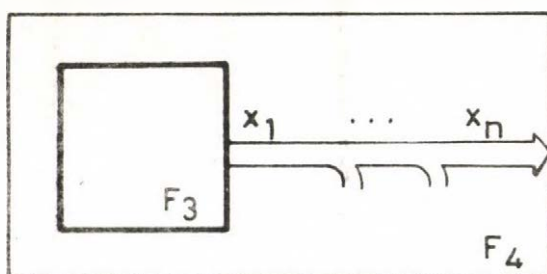
ha $D_1 \setminus D_2$ és
 $-D_1.y_1.y_2. \dots y_m.D_2$ funkcióút,
 akkor $D_1 = O(y_1), D_2 = O(y_m)$;

ha $F_3 // F_4$ és
 $F_3.x_1.x_2. \dots x_n.F_4$ - adatut,
 akkor $x_1 = O(F_3), x_n = O(F_4)$;

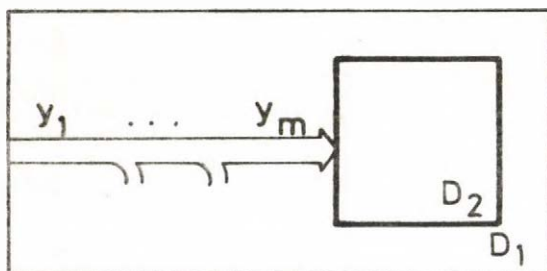
végül ha $D_3 // D_4$ és
 $D_3.y_1.y_2. \dots y_m.D_4$ - funkcióút,
 akkor $D_3 = I(y_1), D_4 = I(y_m)$.



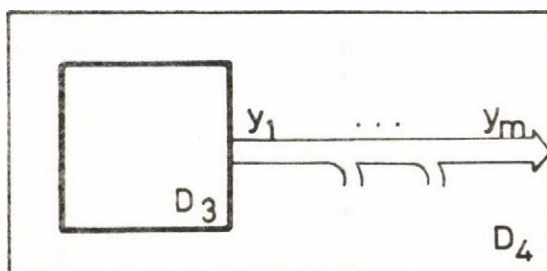
$$-F_1 \cdot x_1 \cdots x_n \cdot F_2$$



$$F_3 \cdot x_1 \cdots x_n \cdot F_4^-$$



$$-D_1 \cdot y_1 \cdots y_m \cdot D_2$$



$$D_3 \cdot y_1 \cdots y_m \cdot D_4^-$$

teljes nyílt adatkapcsolat

$F1.x.F2- \Rightarrow$

155

$x=O(F1)=O(F2)$

Ez az eset csak úgy képzelhető el, ha $F1//F2$;
a kapcsolat "megmagyarázza", hogy $F2$ hogyan
"állítja elő" x -et.

részleges nyílt adatkapcsolat

$F1.x.y.F2- \Rightarrow$

$x=O(F1), y=O(F2), x \& y \neq 0$

Ez az eset részletinformációt hordoz arról,
hogy egy magasabb szintű funkció valamely
kimenete hogyan áll elő részfunkcióinak ki-
meneteiből.

Mindaz, amit az adatutak kapcsán elmondottunk, értelemszerűen
vonatkozik a funkcióutakra és a kapcsolatos I/C/O/M relációk-
ra. Az adatutak bemeneti oldalán (ha szükséges) jelölhető,
hogy vezérlő bemenetről van szó, pl.

$F1.x. \dots y. iF2$ egyszerű bemenet

$F1.x. \dots y.icF2$ vezérlő bemenet

$F1.x. \dots y.imF2$ mechanizmus bemenet

Nyílt adatutak is lehetnek valódiak vagy nem valódiak.

4.2.1.8.3 Nyílt adatút dolgok között

Amint azt 4.2.1.8 bevezetőjének végén említettük, realizáló funkciók és realizált dolgok között is elképzelhető adatút jellegű kapcsolat. A nyílt adatút ilyen adatút egy részét mutatja meg a mechanizmusok (d_i) és a megvalósított dolgok (D_1, \dots, D_2) között.

ha $D_1 \setminus D_2$ és

$-D_1.d_1.d_2. \dots d_n.D_2$ nyílt adatút,

akkor $d_1 = \text{Im}(D_1)$, $d_n = \text{Im}(D_2)$

4.2.1.8.4 Adatutak (funkcióutak) kapcsolása

A nyílt adatutak (funkcióutak) jelentéséből következik, hogy pl.

akár $F_1.x \dots y.F_2- + F_2.y \dots z.F_3$

akár $F_1.x \dots y.F_2 + -F_2.y \dots z.F_3$ helyettesíthető a

$F_1.x \dots y.y \dots z.F_3$ zárt adatúttal.

Vigyázat, $\dots y.y \dots$ nem egyszerűsíthető $\dots y \dots$ -ra, mert ez az utakkal leírt gráf struktúráján módosítana!

Amint azt 4.2.1.8 bevezetőjének végén említettük, realizáló funkciók és realizált dolgok között is elképzelhető adatút jellegű kapcsolat. A zárt adatút explicit módon nem jelenik meg sem aktigramm, sem datagramm modellben, azonban 4.2.1.8.2 és 4.2.1.8.3 szerinti nyílt adatutak 4.2.1.8.4 szerint végrehajtott kapcsoláson keresztül létrejöhet. (A kérdésről ld. még 4.2.1.9.1 alfejezetet.)

4.2.1.8.6 Adatutak (funkcióutak) összefüggései

Két adatút (funkcióút) nem független, ha elejük vagy végük egyezik, t.i. ekkor elágazással vagy csatlakozással van dolgunk. A példa adatút elágazását mutatja:

F1.x.a ... F2 és

F1.x.b ... F3

rövidítése:

F1.x.(a ... F2, c ... F3)

Látható, hogy F1 kimenete x legalább két részre hasad, úgy mint a és b, melyek rendre valamilyen módon F2-höz ill. F3-hoz jutnak el. Amellett, hogy F1 és F2 (ill. F3) közötti összefüggést írtunk le, x,a és b között is főnn kell, hogy álljon az

$x \parallel a$ és $x \parallel b$

reláció, különben $x \&a * x \&b = 0$ lenne, tehát a két adatút

valamelyike nem volna valódi.

4.2.1.9 A funkcionális architektúra modellstruktúrája

4.2.1.9.1 A funkcionális architektúra elemei

Egy rendszer funkcionális architektúráját leírják a funkciókat és dolgokat definiáló fák $\{Tfi\}$, $\{Tdj\}$, az adat és funkcióutak, valamint a struktúra mechanizmus relációk.

Aktigramm modellnek nevezünk egy Tfi definíciós fát, (melynek szögpontjai a funkciók, élei a "része" relációk), a fa szögpontjai közötti kapcsolatot leíró adatutakkal, valamint a fába tartozó és azon kívüli funkciók között kapcsolatot teremtő struktúra mechanizmus relációkkal együtt.

Datagramm modellnek nevezünk egy Tdj definíciós fát, (melynek szögpontjai a dolgok, élei a "része" relációk), a fa elemei közötti kapcsolatot leíró funkcióutakkal, a szögpontok reprezentálta dolgok realizációs mechanizmusait megadó nyílt adatutakkal, valamint a fába tartozó dolgok és azon kívüli dolgok között kapcsolatot teremtő struktúra mechanizmus relációkkal együtt.

4.2.1.9.2 Párbaállíthatósági tétel

Ha vesszük a funkiók és a dolgok definíciós gráfját $(DefF, DefD)$ akkor létezik olyan fába rendezés, hogy az aktigramm és datagramm modellek párbaállíthatóak legyenek. A párbaállítás úgy történik, hogy azok a dolgok kerülnek egy aktigramm párjába, melyek az aktigrammhoz tartozó adatutakban résztvesznek és viszont.

A tétel következménye, hogy egy funkcionális architektúra modellekből áll, melyek mindegyike külön-külön fába rendezett funkiókat és dolgokat ír le, azok kapcsolatával együtt. Egy modell egy aktigramm modellből és a vele párba állított datagramm modellből áll. Az egyes modellek között a (struktúra) mechanizmus reláció teremt kapcsolatot.

Bizonyítás: Tegyük föl, hogy egy aktigramm modellben szerepel egy adat és ugyanennek az adatnak a mechanizmusa is. Ez ugyanis azt jelentené, hogy az adatutak két különböző datagramm modell elemeire hivatkoznának és így nem lehetne a tételben megkívánt párbaállítást egyértelműen elvégezni. (Más esetet nem kell vizsgálni, mert ha a két különböző adat nem azonos datagramm modellhez tartozik ugyan, de nem is áll egyik a másikkal mechanizmuskapcsolatban, akkor a két adatfa egy

közös gyökér alatt egyesíthető volna.)

Elnevezések:

MA egy aktigramm modell, melyben F_l funkciót definiáltuk.

MD egy datagramm modell, melyben D_l dolgot definiáltuk.

mD egy datagramm modell, melyben $dl=M(D_l)$ -et definiáltuk.

Tegyük föl, hogy D_l és F_l között fennáll az I/O/C relációk valamelyike. Ekkor szükséges, hogy dl -nek legyen egy interface funkciója (fl), mely F_l -nek mechanizmusa. Ha mármint dl szerepelne MA-ban, akkor $fl=M(F_l)$ -nek is (mint interface funkciónak) meg kellene jelennie MA-ban. Ez azonban lehetetlen, mert MA definíciószerűen nem tartalmazza semelyik funkciójának mechanizmusát. A párbállítás tehát lehetséges.

Q.E.D.

4.2.2 A funkcionális architektúra szemantikája

Ebben a fejezetben bemutatjuk, hogyan lehet funkcionális specifikációra használni a 4.2.1-ben ismertetett fogalmakat és relációkat.

Ha az ismertetett fogalmakkal és relációkkal fölépítünk egy

architektúrát, akkor bizonyos esetekben szemantikailag definiált lesz a leírt rendszer is. A szemantika kialakulását ebben a fejezetben a funkciók példáján mutatjuk be; azonos szabályok érvényesek a dolgok kombinációira is. A szemantikus tartalmat itt a kialakuló specifikáció jelentése szempontjából vizsgáljuk.

A 4.2.2.1 - 4.2.2.6 pontokban bemutatjuk azokat az alapeseteket, melyek ismert jelentésű részek (pl. funkciók) kombinációjával magasabbrendű funkciókat definiálnak. A bizonyítás módja mindig hasonló: feltételezzük, hogy az alapfunkcióknak van referenciastuktúrája, és megmutatjuk, hogy hogyan konstruálható ebből az újonnan definiált funkció referenciastuktúrája (1) , vagy megmutatjuk, hogy mi a konstruálhatóság feltétele. A tárgyalt változatok:

4.2.2.1 alfejezet : független részek kombinációja

4.2.2.2 alfejezet : definíciós struktúraegyeztetés
alapján

4.2.2.3-6 alfejezet: függőségi megkötések a definícióban

Tegyük föl, hogy ismert F_1 és F_2 , továbbá referenciastuktúra-

(1) A referenciastuktúra mint fogalmi létező és nem mint fizikai struktúra értendő.

ik: $\text{ref}(F_1)$ és $\text{ref}(F_2)$. A belőlük alkotott rendszer nem független részek kombinációja, ha létezik olyan adatút, mely a referenciastruktúrák választását valamilyen módon megköti. A szemantika meghatározottságához a megkötéseket azonossági kapcsolatokra kell visszavezetni. (1)

4.2.2.1 Független részek kombinációja

Ha $F \rightarrow \{F_1, F_2\}$ és nincs F_1 és F_2 között adatút, továbbá F, F_1 ill. F, F_2 közti adatutakban résztvevő dolgok halmaza független, akkor $\text{ref}(F) = \text{ref}(F_1) + \text{ref}(F_2)$. (2)

4.2.2.2 Definíciós struktúraegyeztetés alapján (mechanizmus megadásával)

Ha f -nek ismeretes a referenciastruktúrája és $f = M(F)$, akkor $\text{ref}(F)$ része $\text{ref}(f)$ -nek. T.i. f -nek lehetnek olyan kapcsolatai a mechanizmus szinten, melyeknek F szintjén nincsen megjelenési formájuk. (Pl. egy telefonközpont funkcionális modelljében az egyes funkciók egymással összefüggnek, de egy, a telefon szolgáltatásait használó rendszer szintjén ezek a belső mechanizmuskapcsolatok nem ismertek.) Megfordítva, F -nek csak olyan

(1) Az azonosságot a képletekben "="-vel jelöljük.

(2) Egyesek ilyenkor F -et nem is tekintik rendszernek.

kapcsolatai lehetnek, melyeknek megfelelője f -ben is megtalálható. (1)

4.2.2.3 Teljes adatkapcsolat

Teljes adatkapcsolat esetén (pl. $-F1.x.F2$, vagy $F1.x.F2$), feltéve hogy adott $R1 > \text{ref}(F1)$ és $R2 > \text{ref}(F2)$, $x1 = \text{ref}(X)$ az $R1$ -ben és $x2 = \text{ref}(X)$ az $R2$ -ben, megköveteljük, hogy $x1$ és $x2$ azonos legyen (itt az egyenlőség nem elegendő). Ez az első olyan eset, hogy föl kell hívni a figyelmet arra: noha csak akkor rendelkezik egy rendszerleírás egyértelmű szemantikai tartalommal, ha referenciastuktúrája létezik, a tényleges fizikai realizációnak még számos akadálya lehet. Egyes szakterületeken belül szükség lehet olyan struktúrateervezési módszerekre, melyeket a funkcionális tervezés során betartva a terület realizációs módszerei a gyakorlatban alkalmazhatóak maradnak.

Megj.: $R1 > \text{ref}(F1)$ reláció jelentése = $\text{ref}(F1)$ része $R1$ -nek, tehát $R1$ -nek jegye a $\text{ref}(F1)$ által definiált funkció.
Egyébként $R1$ -nek lehetnek egyéb jegyei is (elláthat más funkciót is).

4.2.2.4 Zárt, részleges adatkapcsolat

Ha zárt, részleges adatkapcsolat köt össze definiált szemantikai funkciót, akkor az adatát mechanizmus szinten azonos

(1) Egy rendszer definiálásának egyértelműsége megköveteli, hogy ha ismeretes $f=M(F)$, akkor $\langle F, Fi \rangle$ relációkat már ne állítsunk föl. V.ö. 4.2.1.7/b. ponttal.

végekkel kell, hogy rendelkezzen (4.2.2.4.1 eset), vagy az adatutak végeinek leszármasztatott funkcióin keresztül kell, hogy az adatút azonos végű adatutakká redukálható legyen (4.2.2.4.2 eset).

4.2.2.4.1 F funkció ($F \rightarrow \{F_1, F_2, F_3\}$) belső kapcsolatai legyenek az alábbi adatutak:

$F_1.A.C.F_2$

$F_1.A.B.F_3$

Ha adott $f_1=M(F_1)$, $f_2=M(F_2)$, $f_3=M(F_3)$, $a=M(A)$, $b=M(B)$ és $c=M(C)$, akkor mechnizmus szinten kell kiderülnie annak, hogyan kapcsolódik B és C az F_1 egyes részeihez. Pl. a mechanizmusmodellben a következő nyílt adatutak adhatják meg a kérdésre a választ:

$f_{11}.b*.a.f_1$ - és

$f_{12}.c*.a.f_1$ -

A referenciamodell megalkothatóságának feltétele ezek szerint: minden olyan R_1 , R_2 , R_3 referenciamodell összege referenciamodellje lesz F-nek is, melyre teljesül, hogy

ha $\text{ref}(M(B)) = \text{ref}(b*)$, $\text{ref}(f_1)$ része R_1 -nek,

$\text{ref}(M(B)) = \text{ref}(b)$, és $\text{ref}(f_2)$ része R_2 -nek

akkor $\text{ref}(b*) == \text{ref}(b)$

és hasonlóképpen

ha $\text{ref}(M(C)) = \text{ref}(c^*)$, $\text{ref}(f1)$ része $R1$ -nek,
 $\text{ref}(M(C)) = \text{ref}(c)$ és $\text{ref}(f3)$ része $R3$ -nak
 akkor $\text{ref}(c^*) == \text{ref}(c)$

Ha a mechanizmus szinten összetett adatot előállító funkció nincs lebontva (nem ismert az adatút folytatása), akkor a szemantika nem jól definiált. Abban az esetben, ha ez a helyzet (pl. egy kommersz berendezés csatlakozó felületéről van szó), nem biztos, hogy fontos annak teljes funkcionális modelljét elkészíteni -- elegendő, ha rendelkezésünkre áll a határfelület (4.2.1.5) definíciója akár funkcionális leírás, akár szöveg formájában.

4.2.2.4.2 Tegyük föl, hogy

$F1.x.y.F2$ és

$F3.z.y.F2$

adatutakon kívül $F2$ leszármaztatott funkciói és $F2$ közötti nyílt adatutak is ismeretesek:

$-F2.y.x.F21$

$-F2.y.z.F22$

Ez utóbbiakat az adatutak 4.2.1.8.4 kapcsolási szabálya szerint egyesítve:

$F1.x.y.y.x.F21$ és

$F3.z.y.y.z.F22$

adatutakat kapjuk. Ezek szerint

$x1=ref(x)$, $x2=ref(x)$, $z1=ref(z)$ és $z2=ref(z)$

jelöléseket téve, ahol

$x1=ref(x)$ és $ref(F1)$ része $R1$ -nek,

$x2=ref(x)$ és $ref(F21)$ része $R21$ -nek,

$z1=ref(z)$ és $ref(F3)$ része $R3$ -nak,

$z2=ref(z)$ és $ref(F22)$ része $R22$ -nek

az $F1, F3, F21$ és $F22$ referenciamodelljére és elemeire az alábbi azonossági relációknak kell teljesülniük, hogy F referenciamodellje ezek egyesítése lehessen:

$x1 == x2$

$z1 == z2$

4.2.2.5 Nyílt, részleges adatkapcsolat

Ha egy funkció (F) és öt definíció részei ($F1, F2$) között nyílt, részleges adatkapcsolat van, pl.

$F1.x.y.F0$ - és

$F2.z.y.F0$ - akkor

a két adatút ábrázolta elágazás miatt következik, hogy

$x//y$ és $z//y$.

Feltéve, hogy ismert

$\{ ref(F1), ref(x) \}$ melyek $R1$ részei

{ $\text{ref}(F2)$, $\text{ref}(z)$ } melyek $R2$ részei

az $R0 \geq \text{ref}(F0)$ létezésének feltétele, hogy $\text{ref}(y)$ -t ismerjük. Ellenkező esetben $F0$ interface-ének referenciája nem volna adott. Ha y -ról egyebet nem tudunk, akkor $y = x \cup z$, így $\text{ref}(y) = \text{ref}(x) \cup \text{ref}(z)$ lenne. Általános esetben azt követeljük meg, hogy $\text{ref}(y) > \{ \text{ref}(x), \text{ref}(z) \}$ legyen, nem zárva ki annak lehetőségét sem, hogy y -nak más részei is legyenek, vagy akár $\langle x, z \rangle$ akár $\langle z, x \rangle$ fennállását.

Megjegyzés: a részleges adatkapcsolatok tárgyalásánál csak azzal az esettel foglalkoztunk, ahol több adatút találkozik egymással. Megállapodás kérdése, hogy formális hibának tekintendő-e másmilyen adatút létezése. Ha egy teljesnek hitt modellt alapulvéve van olyan adatút, melynek két követő eleme különbözik, és nincs másik adatút, mellyel ezen a ponton elágazást, vagy csatlakozást hozna létre, ez úgy tekinthető, mintha a két adatnév szinoním volna, vagy mintha hiányos (hibás) adatnévmegadással állnánk szemben. Gyakorlati analízis esetén föl kell hívni a figyelmet az ilyen pontokra és el kell dönteni, melyik esetről van szó.

4.2.2.6 Közös be- vagy kimenet

Ha két diszjunkt funkció ki- vagy bejövő adatutai találkoznak, az is megkötést jelent az általuk fölépített magasabbszintű funkció referenciamodelljének létezése szempontjából. Erre példa az

$F1.x.y.$

$F2.z.y.$

adatutak, ha $x \& z \neq 0$. Tegyük föl, hogy $x // z$. Ekkor azonos megkötésekkel kell számolni, mint a teljes, vagy részleges adatutakon át (1) létrejött kapcsolat esetében: vagy F2 valamely leszármazottja, vagy valamely mechanizmusnak referenciastuktúrája és F1 referenciastuktúrája között kell az azonossági megkötésnek fennállnia ($x_2 == x_1$).

Magyarázat az alapesetekhez:

Sok esetben a rendszer által felhasznált eszköz (mechanizmus) funkcionális modelljét egy szöveges leírás helyettesíti. Ekkor az eszköz részletfunkcióit összetett adatkapcsolattal felhasználó funkciók (és kapcsolódó funkciók) együttműködése szemantikailag akkor jól definiált, ha fölteszük: a használt leírás szemantikailag definiálta az eszközt, és eszerint struktúráltta a tervező az éppen készített rendszert. Ezt az esetet -- praktikuma mellett -- azért is érdemes vizsgálni, mert nemcsak az lehet az analízis tárgya, hogy egy rendszerterv teljes-e, hanem az is, hogy megtudjuk, minek föltételezésével tekinthetünk egy rendszertervet teljesnek (és így szemantikailag jól definiáltnak a rendszert).

4.2.2.7 Kapcsolat a megvalósító mechanizmuson keresztül

A megvalósító mechanizmusra nézve mindaz fennáll, amit a közös bemenetekre mondtunk (4.2.2.6). Ezenkívül a megvalósító mechanizmus jelentésmagyarázó szerepéről az alábbiak mondhatók el. 4.2.2.7.1 a funkciófa levelének esetét tárgyalja, mégpedig (a.) annak jelentésadó szerepét, (b.) a struktúra és a megvalósító mechanizmus ellentmondásmentességi

(1) ld. 4.2.2.3, 4.2.2.4, 4.2.2.5

kritériumát. 4.2.2.7.2 a nem levél funkciók mechanizmusairól szól: (c.) az egyszerű bemenetközösség esetét tárgyalja, (d.) pedig azt az esetet, amikor a jelentésadó szerep feltételes, és csak külön diszciplínáris alapokon bizonyítható.

4.2.2.7.1. Egy funkciófa egyik levelének realizációja

Mivel itt nem ismeretes a levélfunkció (pl. F) egy része sem, $k=1$ kell, hogy legyen (v.ö. 4.2.1.4.2.)

a./ Ha $M(F)$ nem ismert, akkor az $R(F)$ realizáció a definíciós struktúra szerepét veszi át.

b./ Ha $M(F)$ ismeretes, akkor $R(F) = \text{Im}(M(F))$
 Szavakban: F megvalósítása mechanizmusnak egyik megvalósításával azonos. Ezekszerint -- bárha $M(F)$ egyedi -- a FA megvalósításakor F egyszerre több példányban, sőt többféle elven is megvalósítható.

4.2.2.7.2 Egy nem-levél funkció (F) megvalósítása

c./ Egyszerűbb esetben $F = \bigcup F_i$ és $\{R(F_i)\}$ ismert. Eszerint a definícióban leírt eset áll fenn: a megkötések figyelembevételével az egyes realizációk egyesítése az egyesített funkció

realizációja. Az egyesítésnek név is adható, és minthogy $\{R(F_i)\}$ elemei és $R(F)$ mindannyian dolgok, dologi struktúrájuk is tárgyalható (része és \parallel relációk, továbbá I/C/O relációk).

d./ Előfordul, hogy az összetett funkciót megvalósító fizikai rendszer nem térben partícionálható (funkciói szerint). Ekkor F (és részei) megvalósításához egyetlen fizikai rendszert rendelünk, továbbá egy diszciplína bizonyítékát arról, hogy a fizikai rendszer a kívánt funkcióegyüttest hordozza.

4.2.3 A FA teljessége és ellentmondásmentessége

Ez a fejezet a FA teljességének és ellentmondásmentességének kérdésével foglalkozik. A kritériumokat (az eddig követett gyakorlat szerint) a funkciók nézőpontjából modjuk ki, de a dologi nézőpont szerinti megfogalmazás éppoly jogosult. Mivel vannak esetek, amikor nem dönthető el egyértelműen, hogy ellentmondás vagy hiány a hiba oka, ilyen bontást a 4.2.3 fejezetben nem alkalmazunk.

4.2.3.1 Interface szabály

Ha egy funkció kapcsolatait leíró nyílt és zárt adatutakat összevetjük, látható, hogy bennük redundancia áll fenn. Ez

ellentmondás forrása lehet. Ha létezik

F.x

adatút, akkor kell, hogy létezzen a következő adatutak közül pontosan az egyik:

..... .x.F- vagy

M(F).M(x).

A második sorra, mint egy másik modell egy adatútjára (azon a modellen belül) szintén érvényes a szabály.

Magyarázat: ha egy funkció valamit használ, vagy előállít, az mindenképpen részei együttműködésének eredménye (vagy mechanizmusa részeiének). Speciális, a fentieknek ellentmondó adatút az -F....F- alakú, melyet formálisan kizárunk. Az a funkció (és dolog), amelynek sem része, sem mechanizmusának része nem ismeretes, terminális. A rendszerleírás terminális elemeken nyugszik, és -- a kapcsolati megkötések betartása esetén -- szemantikája olyan mértékben adott, amennyire terminális elemei.

A terminális elemek szemantikája megadható valamely referenciamodellel, lett legyen az szöveges leírás vagy identifikálható megvalósítás. Nem egy esetben viszonylag bonyolult rendszerek architektúrája közölhető nagyon egyszerűen, főltéve, hogy a közlésben résztvevők birtokában vannak a megfelelő magasszintű

referenciamodelleknek.

Egy-egy műszaki újdonság, ötlet alapja az, hogy eredetileg össze nem tartozó, de a szemantikailag helyes kombináció szabályait kielégítő referenciamodellekből egy új, még nem létező rendszer referenciamodelljét alkotjuk meg (t.i. ekkor van egyáltalán precíz értelemben vett "jelentése" az ötlet közlésének). Ez a kombináció történhet véletlenül, vagy egy kívánt cél elérése céljából [YOS81].

4.2.3.2 Része relációk és adatutak ellentmondása

Az $1 \rightarrow n$ és $n \rightarrow 1$ elágazásai "leszármazottja" ($//, \backslash$) relációt generálnak. Az $n-n$ elágazások, bár nem helytelenek, nem generálnak leszármazottja relációt (a SADT diagrammokon ezt a típust nem szokás megengedni).

Képezzük az összes leszármazottja relációt! Pl. ha

Fi1 . alfa . a . b . beta1 . Fj1

Fi2 . alfa . a . c . beta2 . Fj2

akkor a közös kezdetű adatutak a szétválasztás helyén $1 \rightarrow n$ típusú elágazást generálnak, ami az $a \backslash b$ és $a \backslash c$ relációkat implikálja. A közös végű utak hasonlóképpen $n \rightarrow 1$ típusú csatlakozást írnak le, pl. ha

$Fi1 . betal . b . a . Fj1$ és

$Fi2 . beta2 . c . a . Fj2$

akkor $b//a$ és $c//a$.

Mivel a része relációk fát alkotnak, a leszármazottja reláció pedig e fán belüli lehetséges utak eleje és vége között állhat csak fön, az aktigrammokon definiált leszármazottja relációkat ábrázoló gráf $Ld = \{U \quad Fi \setminus Fj\}$ minden éle származtatható kell, hogy legyen a datagrammokon leírt része relációk $Rd = \{U \langle Di, Dj \rangle\}$ fájából. Ha az Rd -ből származtatható összes leszármazottja reláció tranzitív lezárását $TR \quad Rd$ -vel jelöljük, akkor az aktigrammok hibás (datagrammal ellentmondó) elágazásai:

$$\{hibahelyek1\} = Ld \setminus TR \quad Rd$$

hasonlóképpen a datagrammstruktúrának az aktigrammokban ellentmondó helyek:

$$\{hibahelyek2\} = Lf \setminus TR \quad Rf$$

4.2.3.3 Fába rendezhetőség

Az összes dolog illetve funkció modellenként fába rendezhető legyen. Ez a hiba még akkor is főléphet, ha nincs ellentmondás a része relációval (pl. nem is készült datagram és így nincs is része reláció explicite definiálva).

A 4.2.1.9.2 tétel szerint ugyanis a fába nem illő dolgok nem szerepelhetnek az aktigrammokhoz tartozó adatutakban.

4.2.3.4 Unicitási követelmény

(Sok esetben ellentmondás formájában jelentkezik ez a hiányosság, és csak akkor dönthető el, hogy hiba vagy hiány volt az okozója, ha a vizsgálat kezdetén feltesszük: teljes a ter-
vünk.)

A modellekben a legalsó szintű adatok egyediségét biztosítani kell, nem lehet két azonos nevű dolog (funkció) megkülönböztethető. Ezért adatutakra igaz, hogy ha

minden F_{ii} -re $x = I(F_{ii})$ és

minden F_{Oj} -re $x = O(F_{Oj})$ akkor

minden F_{ii}, F_{Oj} párhoz létezik adatút, hogy

$F_{Oj}.x \dots x.F_{ii}$,

egyébként lenne legalább egy olyan adatút, pl.

$F_1.x \dots x.F_2$,

melyen keresztül x csak F_1 és F_2 között teremt kapcsolatot és ez nem ugyanaz az x , ami a többi F_{Oj} és F_{ii} közötti kapcsolatban szerepel. Ebből az következik, hogy x nem terminális (tovább bontható), tehát teljességi hiánnyal állunk szemben, vagy ha terminálisnak szántuk, akkor hibás a lebon-

tds.

4.2.3.5 Határfelületi (interface) struktúrák egyezése

Az adatutak összes lehetséges összevondását elvégezve, majd véve ebből a terminális funkciókat összekötőket, az összes valódi adatút eleje és vége meg kell, hogy egyezzen. (Hasonlóképpen a közös be- és kimenetekre.) Ez a teljességi kritérium azért kell, hogy fennálljon, mert csak ebben az esetben vezethetjük vissza az adatutakon keresztüli kapcsolatokat azonossági megkötésekre. Az egyezés nem feltétlenül egy modellen belül jön létre, az adatutakat a teljes FA-n keresztül kell összevonni (ld. 4.2.2.4, 4.2.2.5, 4.2.2.6).

4.2.3.6 Tétel

Ha a funkcionális architektúra az 4.2.3.1,-.2,-.3,-.4,-.5 értelmében teljes és ellentmondásmentes, és a terminális dolgok és funkciók szemantikailag definiáltak, akkor a rendszerfunkció jelentése definiált.

Bizonyítás: A referenciamodell létezésének szükséges és elégséges feltétele, hogy

- a./ azonos nevű fogalmak azonos referenciával rendelkezzenek,
- b./ a legalsó szintű dolgok és funkciók referenciája denotáció útján adott legyen.

Elegendőség: A b./ követelmény teljesül, ez a kiindulás (a terminális fogalmak szemantikája a 4.2.3.1 szerint adott kell legyen). A teljességi követelmények (4.2.3.4, 4.2.3.5) miatt a legalsó szintű dolgok és funkciók definíciója egyedi lesz, így az a./ követelmény is teljesül. A fába rendezhetőség követelményei (4.2.3.2, 4.2.3.3) szerint a nem terminális fogalmak mind fölépíthetők a 4.2.2 szerinti kombinációkból, következésképpen referenciastruktúrájuk is létezik.Q.E.D.

Szükségesség: 4.2.3.2 és 4.2.3.3 bármelyikének megsértése esetén nem volna egyértelmű fába rendezése a fogalmaknak, tehát a 4.2.2 szerinti kombinációknak (melyeket a fába rendezhetőség föltételezésével alkottunk) nem volna továbbá egyértelmű jelentésük, hiszen definíciójukban szerepel a része reláció. Természetesen ez nem jelenti azt, hogy másfajta rendszert, amely a szemantika megadását ettől eltérő alapokon oldja meg, nem lehetne konstruálni. A 4.2.3.4 és 4.2.3.5

teljesülése azért szükséges, mert ha nem kötjük ki az egyediség követelményét, nem lehet megállapítani, hogy a funkcionális architektúra készítője ténylegesen azonos, vagy csak véletlenül egyező fogalomneveket használt-e, tehát többfajta értelmezésre nyílna lehetőség. 4.2.3.1 pedig szükséges, ellenkező esetben a felsőbb szintű fogalmak terminálisra visszavezetése hiányozna a funkcionális architektúrából, így volna legalább egy olyan funkció vagy dolog, melynek nincs referenciamodellje. Q.E.D.

A 4.2.3.1-6 követelmények tehát a 4.2.1 szerinti alapfogalmak és relációk segítségével felépített funkcionális architektúra szemantikailag definiált (jelentéssel bíró) voltának szükséges és elégséges feltételét adják.

4.2.4 Eszközülasztás és kísérletek az SDLA rendszerrel

A nagyméretű funkcionális tervek készítését két ponton is kívánatos automatikus eszközökkel támogatni.

a./ A terv készítése közben a tervezőt el kell látni a tervezési folyamatot segítő, annak megfelelő formába rendezett információval.

b./ Az elkészült terveket értékelni kell.

Mindkét feladat azonos analízislépéseken nyugszik, de ezeket más-más kombinációban kell végrehajtani. A tervezés folyamán, amíg a terv még nem teljes, sokszor nem lehet egyértelműen eldönteni: hibás-e vagy csak hiányos. Nincs szükség továbbá arra, hogy minden alkalommal a teljes, már elkészült rendszertervet ellenőrizzük. Ehelyett lehetőséget kell adni a terv részenkénti analízisére is. Ezekhez a feladatokhoz járul hozzá az ISDOS rendszerrel kiépített kapcsolat, és eszközeiben ilyen rendszer kísérleti megalapozását írja le ez a fejezet is.

Az elkészültnek nyilvánított rendszertervek analízisének eredménye más szempont szerint nyújtja ugyanazt az információt, mint a menet közben végzett analízis. Pl. ki kell gyűjteni minden olyan terminális részletet, melynek ismerete a rendszerterv megértésének alapfeltétele (kész terv esetén), vagy meg kell keresni mindazt az alsószintű fogalmat, melynek további bontását el kell végezni (menet közben végzett analízis).

4.2.4.1 Eszközválasztás

Ahhoz, hogy a 4.2.3-ban megfogalmazott analízist elvégezhessük, olyan adatbázis kell, melyben tárolni lehet a 4.2.1-ben leírt relációkat. Ezen túlmenően az analízisfeladatokat a

tervezési folyamatba kell helyezni, így tárolni kell az ezzel kapcsolatos járulékos információt is (pl. ki, mikor, hol, minek alapján állította egy reláció fennállását).

Ez utóbbi csoportra igen nagy szükség van, mert a tervezés segítéséhez nemcsak a terv helyes vagy helytelen voltát kell kimutatni, hanem a hiba lehetséges forrását is, továbbá föl kell fedni a tervezők közötti ellentmondásokat is.

A leírt analízislehetőségek mindannyian gráfok vizsgálatára vonatkoznak. Ezek a gráfok ábrázolják a

- része relációkat
- leszármazottja relációkat
- adatút és funkciót hálózatokat
- a modellek közötti kapcsolatokat rögzítő mechanizmus-relációkat.

A gráfokat relációalgebrai eszközökkel leírva, majd bővítve a relációkat a tervezési folyamatra vonatkozó attribútumokkal, előáll a tervezési adatbázis tartalma. Külön követelmény, hogy az "adat" és "reláció" fogalmát lehetőleg ne válasszuk külön egymástól, mivel a tervezési folyamat során az egyes tervezési funkciók kimenő adatai a tervezett rendszert leíró relációk. Így kétséggkívül szerencsés dolog az ábrázoláshoz referencia típusú relációs adatbázist használni. Bár az ábrázolás más módon is megoldható, ez tükrözi leginkább a természetes adatstruktúrát [JAC75], tehát adekvát eszköz.

A relációs adatbázisok e típusának képviselője az SDLA rendszer. Az SDLA túl ezen az alapvető tulajdonságon rendelkezik egy meta szintű bemeneti nyelvvel, mely a fogalmi struktúra leírásának eszköze. A meta szinten definiált fogalmak (relációk) példányainak létrehozása felhasználói (írott szöveg formájú) csatlakozó felületen keresztül történhet.

A funkcionális modellek tárolására és analízisére elvben bármely relációs adatbázis megfelel. Az SDLA választását két körülmény határozta meg:

hozzáférhetőség

a referencia típusú relációs adatbázis "természetes" képessége arra, hogy a relációegyedeket adatként ábrázolja (pl. névadás lehetősége).

A feladat megvalósíthatóságának demonstrálására kísérleti kapcsolat létesült a grafikus modellező rendszer és az SDLA (CDC 3300-as implementációjú) teszt változata között.

A kapcsolat létrehozásának lépései az alábbiak voltak:

a.) megfogalmaztuk a funkcionális modell relációit az

SDLA meta nyelvén

- b.) elvégeztük egy kisméretű modell leírását
- c.) példaként kialakítottuk az analízislépések egy megfogalmazását.

A II. függelékben található metanyelvi megfogalmazás már nemcsak a funkcionális modellek információtartalmát írja le, hanem annak grafikus reprezentációjához kötődő (a tervezési folyamatra utaló) adatokat is. Így ugyanis -- az információ keletkezési helyének tárolásával -- a tervben talált ellentmondások felfedésekor azonnal megkapható, hogy a grafikus reprezentációban mely ábrák mely pontjai mondanak ellent egymásnak.

4.2.4.2 Az analízislépések végrehajtása

A relációs adatbázis tartalmának vizsgálatára két lehetőség van. Az első az analízis eredményformátumának megadása relációkifejezésként, melynek közvetlen kiértékelése az analízis végrehajtását jelenti. Ezt az utat követtük a II. függelékben közölt egyszerű példában is. A példa a "része" és "leszármazottja" relációk ellentmondásának kiértékelését mu-

tatja be. Itt is látható, hogy a "része" reláció tranzitív lezárásnak leírása relációkifejezésekkel eléggé nehézkes, továbbá a közbenső tárolás hiánya miatt igen időigényes is.

A második út, a relációs adatbázis lekérdezése egy algoritmus nyelven írt program segítségével, az előbbi ok miatt gyakorlati méretű feladatok esetén célszerűbb. Az SDLA rendszer meta szinten megengedi, hogy a definiált relációra nézve strukturális kritériumokat kössünk ki. Ezek teljesülésének hiánya az adatdefiníciós részek visszautasítását vonja maga után. Emiatt az SDLA ezen lehetőségére nem támaszkodtunk, mert átmenetileg hibás tervek tárolását is meg akartuk oldani -- az SDLA-t mint tervezési adatbázist használva.

A gyártórendszereknél szükséges tervméreteket esetén nem engedhető meg a teljes tervezési adatbázis állandó újragenerálása. Ezért, ha az SDLA rendszer és a grafikus modellező rendszer kapcsolatát üzemszerűen (elfogadható költségekkel működtethető termék szintjén) kívánnánk kiépíteni, nem volna elegendő a grafikus modellek egyirányú lefordítása a meta szint által definiált adatnyelvre és ennek automatikus adatbázisba vétele, hanem szükség volna az adatbázissal kapcsolatos összes gyakori művelet program-interfacen keresztüli elérésére is (u.m. részleges visszakeresés, írás, olvasás, törlés, rendezés stb.).

Az itt leírt kísérletek és meggondolások csupán elvi alapot szolgáltatnak egy termék szintű tervezőrendszer kialakításához.

5. Összefoglalás

Ebben az értekezésben az 1976-83 között végzett kutatómunkám eredményét foglaltam össze. Egyéni kutatómunkám mellett – amint az irodalomjegyzék is tanúsítja – az eredmények team munkában végzett kísérletekhez, alkalmazásokhoz és fejlesztésekhez is kapcsolódnak.

Az itt összefoglaltak közül legfontosabbnak azt tartom, hogy a fent nevezett időszakban kialakult egy módszertan,

- mely nemzetközileg nem elszigetelt unikum, ugyanakkor saját és kézbentartott eredmény, és
- a gyakorlat elméleti problémáiból kiindulva tudományos alapokra helyezi nagy gépipari rendszerek funkcionális tervezését.

A gyakorlati munka folytatásaként (melyről ez az értekezés már nem számolhatott be részletesen) több olyan kutatási és fejlesztési munka indult, mely támaszkodni kíván a kidolgozott módszertanra. (A módszertannak utólag a SATT¹ nevet adtuk.) Így pl. két hazai szerszámgépgyár által létesítendő összesen hat gyártórendszer rendszertervezési munkái várhatóan ezzel a módszerrel fognak készülni. A következő ötéves tervre előirányzott IAAR kutatási munkák tematikája hasonló (de nem közvetlenül termék kifejlesztésére irányuló) rendszertervezési munkákat irányoz elő.

1 SATT= Struktúrált Analízis és Tervezési Technika
(Structured Analysis Techniques and Technology)

A kidolgozott számítógépes eszközök és a formalizálási eredmények ma reálissá teszik egy olyan termék kifejlesztését, mely egy tervezőiroda kezébe adva gyártórendszerek ajánlati dokumentációjának és rendszertervének számítógéppel segített előállítására ad módot. A számítógépes támogatás és a módszertan további fejlesztése két olyan téma, melyben nemzetközi együttműködést tartunk fenn (Polytechnic of the South Bank, London; Laboratoire G.R.A.I., Université de Bordeaux I.).

A rendszertervek szemantikai ellenőrzésére bemutatott eredmények részint az előbbi bekezdésben említett fejlesztésekben hasznosulhatnak, részint a további kutatómunka alapjául szolgálhatnak. Két kutatási irány ma igen aktuális és ígéretesnek is tűnik:

- a.) A logika és a funkcionális analízis közötti összefüggés tudományos rendszerbe foglalása (részint a tervezési módszertanra visszahat, részben a realizációs módszerekkel kivitelezhető kapcsolat a várományos felhasználója ilyen típusú eredményeknek).
- b.) A funkcionális architektúra és teljes rendszertervek számítógéppel segített szemantikai analízise. A feladat a kidolgozott elmélet szerint referenciamodellek ismeretében oldható meg. Ez a tudásreprezentációs módszerek, szakértő rendszerek eszköztárának mozgósítását indokolja. Végössorón a mesterséges intelligencia egyik várományos felhasználója a számítógéppel segített rendszertervezés. Véleményem szerint a jövőben ez a terület a természetes nyelv megértéséhez hasonló alapokra helyezhető, hiszen a funkcionális

analízis referenciamodell fogalma a természetes nyelvek megértésének kulcslépésében kialakuló belső szemantikai modellével rokon.

IRODALMI HIVATKOZÁSOK

- [ADE79] -,
Le GRAFCET, diagramme fonctionnel des automatismes
sequentiels.
ADEPA, Paris, (Avril 1979) p.22.
- [ADE80] -,
GEMMA - Guide d' étude des modes de marches et d'
arrêts.
ADEPA, Paris-Montrouge (1980) p.32.
- [AF 78] ,
Integrated Computer Aided Manufacturing - ICAM Program
Prospectus.
Air Force Materials Laboratory W.P.A.F.B (June 1978).
- [BAN80] Bandman, O.L.,
Szintez aszinhronnovo mikroprogramnovo upravlenia pa-
rallelnümi processzami,
Kibernetika, No 1, (1980) pp.42-47.
- [BAS75] Bastarche, M.J, E.A. Hershey III,
Problem statement analyser command descriptions.
ISDOS W.P. No 91, Dept. of Industrial Eng. and
Operations Eng., Univ of Michigan, Chicago, ILL, (March
1975)
- [BER68] von Bertalanffy, L.,
General System Theory.
George Braziller Inc., NY, NY (1968)
- [BER78] Bernus, P.,
Az SADT elnevezési konvencióiról (Belső leírás).
MTA-SzTAKI, Budapest, (1978 május) p.25.
- [BER79a] Bernus, P.,
A multitask Supervisor for Intel-8080 Microprocessors.
Proc. Winter School on the Theory of Operating Sys-
tems, Visegrád, Tanulmányok 100/1979, MTA-SzTAKI, Bu-
dapest, (January 1979) pp.45-63.
- [BER79b] Bernus, P., J. Hatvany,
Computer Aided Design of Integrated Manufacturing
Systems.
Computers in Industry, Vol.1., No.1., (June 1979)
pp.11-19.

- [BER80] Bernus P., Kovács V., Vészi Á.,
Az IGYR-630 rendszerterve.
MTA-SzTAKI, Budapest, (1980 június) p. 100 /kb./
- [BER81a] Bernus, P.,
Gondolatok a gépipar termelőrendszereiről.
Working Paper GCs/2, MTA-SzTAKI, Budapest, (1981 június) p.21.
- [BER81b] Bernus, P., J. Hatvany, L. Nemes,
Computer Aided Design of Integrated Manufacturing
Systems - Some Experimental Results.
IFAC VIII. World Congr., Kyoto, Japan, Prepr. Vol.XIV.
(August 1981) pp.135-140.
- [BER81c] Bernus P., Dr. Gerencsér P., Kovács V., Vészi Á.,
Komplex gépipari rendszerek funkcionális tervezési
módszertana /Kézikönyv/.
MTA SzTAKI, Budapest (1981 szeptember) p.143.
- [BER81d] Bernus P., Dr. Gerencsér P., Kovács V., Vészi Á.,
Komplex gépipari rendszerek funkcionális tervezési
módszertana /Tervezői kézikönyv/.
MTA SzTAKI, Budapest (1981 szeptember) p.377.
- [BER81e] SzTAKI-SzIMFI tervezőcsoport (Bernus P. SzTAKI témavezetőként),
Felügyelet nélküli gyártócella szekrényes alkatrészek
megmunkálására /követelményrendszer/.
MTA-SzTAKI, Budapest/ SzIMFI, Halásztelek (1981 október) p.150.
- [BER82a] Bernus P.,
Gyártórendszerek és gyártócellák megvalósulási folyamata.
Working Paper GCs/3, MTA-SzTAKI, Budapest, (1982 június) p.48.
- [BER82b] Bernus P., Kovács V.,
TPA/70 - VT20/A kapcsolat.
MTA-SzTAKI/GAFO (Belső leírás), (1982 május).
- [BER82c] Dr. Gerencsér P, Bernus P., Vészi Á.,
Interdisciplinary Teamwork for Large Scale Systems
Design.
Working Paper GCs/4, MTA-SzTAKI, Budapest, (1982 augusztus) p.9.

- [BER82d] SzTAKI-SzIMFI tervezőcsoport (Bernus P. SzTAKI témavezetőként),
Felügyelet nélküli gyártócella szekrényes alkatrészek megmunkálására /Funkcionális rendszerterv/.
RT-N3/82-Sz649, MTA-SzTAKI, Budapest/ SzIMFI, Halásztelek, (1982 november) p.150.
- [BER83a] Bernus P., Dr. Gerencsér P., Kovács V., Vészi Á.,
Szerszám és készülék információs rendszer gyártórendszerek számára /Funkcionális rendszerterv/.
RT-N36/83-Sz679, Working Paper GCs/11, MTA SzTAKI, Budapest, (1983) p.94.
- [BER83b] Bernus P., Kovács V.,
Szerszám és készülék információs rendszer gyártórendszerek számára /Megvalósítás rendszerterve/.
RT-N11/83, Working Paper GCs/9, MTA SzTAKI, Budapest, (1983) p.18.
- [BER83c] Kovács V., Bernus P.,
Szerszám és készülék információs rendszer gyártórendszerek számára /Kezelési útmutató/.
SD-N4/83, MTA SzTAKI, Budapest, (1983) p.41.
- [BER83d] Bernus, P.,
Rigour and Permissivness in the Design of CAD/CAM Systems - Theory and Practice of a Methodology.
Preprints of the IFIP W.G. 5.2/5.3 Working Conference on the Integration of CAD/CAM, Gaussig, GDR (1983) pp.1-21.,
(Also EK-N42/83, Working Paper GCs/6, MTA SzTAKI, Budapest, (1983) p.22.)
- [BIE78] Biewald, J., P. Goehner,
EPOS - a Specification and Design Technique for Computer Controlled Real-time Automation Systems.
Proc. of 4-th Int. Conf. on Softw. Eng., Munich, (September 1978) pp.245-250.
- [CAM75] CAM-I Advanced Technical Planning Committee,
A Long Range Plan for Community Development of Information Services, to Improve Productivity in Design and Manufacturing.
CAM-I, (1975) p.43.
- [COC81] Committee on Computer Aided Manufacturing,
Report on Activities January 1980 to June 1981.
National Academy Press, Washington D.C. (1981) p41.

- [CZU82] Czulek A.,
Programrendszer vegyipari folyamatok időbeli működésé-
nek vizsgálatára.
Automatizálás, 82/6 (1982) pp.11-39.
- [DEM82] Demetrovics, J., E. Knuth, P. Radó,
Specification Meta Systems.
COMPUTER (May 1982) pp.29-35.
- [DIC78] Dickover, M.E., Cl.L. McGowan, D.T. Ross,
Software Design Using SADT.
Infotech State of the Art Report on Structured Analy-
sis and Design, (1978) pp.99-113.
- [FEI82] Feiner, S., S.S. Nagy, A. van Dam,
An Experimental System for Creating and Presenting
Interactive Graphical Documents.
ACM Trans. on Graphics, Vol.1., No.1., (1982)
pp.59-77.
- [GAN79] Gane, C., T. Sarson,
Structured Systems Analysis: Tools and Techniques.
Prentice Hall, Englewood Cliffs, NJ, USA, (1979).
- [HAT77] Hatvany, J.,
Contribution to the Discussion of Interactive Command
Languages in CAD.
In J.J. Allan (Ed) CAD Systems, North Holland,
Amsterdam (1977) p.215.
- [HAT78] Hatvany, J., P. Bernus,
An Approach to the Design of Large CAD/CAM Systems.
Presented at IFIP/CIRP/CMEA/CAM-I Joint Session,
Bournemouth, England, (February 1978).
- [HOR72] Hori, S.,
The Structure of Functions and its Application to CAM
Planning.
NC Scene, (July 1972) pp.2-5.
- [JAC75] Jackson, M.A.,
Principles of Program Design.
Acad. Press. New York (1975)
- [KNU79] Knuth, E., P. Radó, A. Tóth,
Preliminary description of SDLA.
MTA-SzTAKI Tanulmányok 105/80, Budapest, (December
1979)

(magyar nyelven is: Tanulmányok 104/80)

- [KNU80] Knuth, E., P. Radó,
Principles of computer aided system description.
MTA SzTAKI Tanulmányok 117/81 (November 1980)
- [KNU82] Knuth, E., F. Halász, P. Radó,
SDLA System Descriptor and Logical Analyser.
In T.W. Olle, H.G. Sol, A.A. Verrijn-Stuart (Eds) Proc.
of the IFIP Comparative Review on Information Systems
Design Methodologies, North Holland, Amsterdam, (1982)
pp.143-177.
- [KOV82] Kovács V.,
A funkcionális tervezés, mint nagy rendszerek megbíz-
hatósága növelésének egyik biztosítója.
Proc. RELECTRONIC'82, Budapest, OMIKK Technoimpex,
Budapest, Vol.I. (1982) pp.135-144.
- [KOV83] Kovács V., Halász M.F.,
Szerszám és készülék információs rendszer gyártórend-
szerek számára /Software követési dokumentáció/.
KL-N10/83-SZ679, Working Paper GCs/8, MTA SzTAKI,
Budapest, (1983) p.56.
- [LAD81] Ladó L., Náhlik G.,
A módszeres innovációs diagnosztika.
Vezetéstudomány, XII/5 (1981) pp.7-14.
- [LAD82] Ladó, L.,
Az alkotásra való felkészítésről.
BME Ipari Üzemgazdaságtan Tanszék, Budapest, (1982).
- [LIP76] Lipton, R.,
The Reachability Problem and the Boundedness Problem
for Petri-nets are Exponential-space Hard.
TR-62, Computer Science Dept., Yale University,
(January 1976).
- [MAR78] DeMarco, T.,
Structured Systems Analysis and System Specification.
Yourdon Inc., New York, N.Y., USA, (1978).
- [MEK80] Mekly, L.J., S.S. Yan,
Software design representation using abstract process
networks.
IEEE Trans. on Software Engineering, Vol.SE-6, No.5,
(September 1980) pp.420-435.

- [MOA81] Moalla, M., R. David,
Extension du GRAFCET pour la représentation de
systemes temps réels complexes.
R.A.I.R.O. Automatique, Systems Analysis and Control,
Vol.15, No.2, (1981) pp.159-192.
- [MUL78] Mullery, G.P.,
CORE - a Method for Controlled Requirement Specifica-
tion.
Proc. of 4th International Conf. on Software Eng.,
Munich, FRG, (September 1978) pp.126-135.
- [MUN83] Munro, A.T.D.,
SADIST: an Interactive editor for Structured Analysis.
Computer Graphics Forum, Vol.2, No.2/3, (August 1983)
pp.104-114.
- [NOE78] Noe, J.D.,
Hierarchical Modeling with Pro-nets.
In W.R.Tranta (Ed.) Proc. of Nat. Electr. Conf.,
Vol.32, Chicago, ILL, (October 1978) pp.155-160.
- [PAL78] Palkó B.,
IKARUS robottechnika (Felmérő tanulmány).
II. Kötet., GAFO-AMASZO, MTA-SzTAKI, Budapest, (1978)
p.177.
- [PAL79] Palkó B., Bernus P., Palkó L.,
Strukturált Analízis és Tervezési Módszer.
Automatizálás XII.Évf., 10.sz., (1979 október)
pp.2-12.
- [PAR72] Parnas, D.L.,
On the Criteria to be Used in Decomposing Systems into
Modules.
Communications of the ACM, (December 1972).
- [PAR78] Parnas, D.L.,
Some Software Engineering Principles.
Infotech State of the Art Report on Structured Analy-
sis and Design, Vol.2., (1978) pp.237-247.
- [PET78] Peterson, J.L.,
An introduction to Petri nets.
In W.R.Tranta (Ed.) Proc. of Nat. Electr. Conf., Vol
32, Chicago, ILL, (October 1978) pp.144-148.

- [RAD80a] Radó, P., O. Kiss, E. Knuth, A. Szilléry,
SDLA 1.0 Users Manual.
W.P. No II/14, MTA SZTAKI, Budapest, (November 1980).
- [RAD80b] Radó P., Kiss O., Szilléry A.,
Relációs adatbázis interface.
W.P. No II/10, MTA SZTAKI, Budapest, (1980 Augusztus).
- [RAM80] Ramamorthy, C.V., G.S. Ho,
Performance Evaluation of Asynchronous Concurrent
Systems Using Petri-nets.
IEEE Trans. on Software Engineering, Vol. SE-6 No. 5,
(September 1980) pp. 440-449.
- [ROS77] Ross, D.T.,
Structured Analysis (SA): a Language for Communicating
Ideas.
IEEE Trans on Softw. Eng., Vol. SE-3, No. 1, (January
1977) pp. 16-34.
- [SCH78] Schoman Jr., K.,
SADT and Simulation.
TP072, SofTech Inc., Waltham, MA, USA, (1978) p. 20.
- [SOF76] -,
An Introduction to SADT, Structured Analysis and
Design Technique.
No. 9022-78, SofTech Inc., Waltham, MA, USA, (February
1976) p. 23.
- [STA76] Stay, J.,
HIPO: an Integrated Program Design.
IBM Systems Journal, 15, 2, (1976) pp. 143-154.
- [STE74] Stevens, W., G. Myers, L. Constantine,
Structured Design.
IBM Systems Journal, 13, 1, (1974) pp. 114-139.
- [SZE79] Szeredi P.,
Az LDM tervezési nyelv és módszer.
I., II., III. Köt., SZÁMKI, Budapest, (1979).
- [TEI74] Teichrow, D., E.A. Hershey III, M.J. Bastarche,
Information Systems Description Analysis, an introduc-
tion to PSL/PSA.
ISDOS W.P. No 86, Dept. of Industrial Eng. and
Operations Eng., Univ of Michigan, Chicago, ILL, USA,
(March 1974).

- [TEI75] Teichroew, D., M.J. Bastarche,
PSL users manual.
ISDOS W.P. No 98, Dept. of Industrial Eng. and Operations Eng., Univ of Michigan, Chicago, ILL, USA, (March 1975)
- [TEI81] Teichroew, D.,
The development of software support environments.
AGARD Symposium on Airborne Distributed Computing and Networks, (June 1981) p.14.
- [TRA76] -,
TRAIDEX Needs and Implementation Study.
SoftTech Inc., Waltham, MA, USA, (1976) p.150 (approx).
- [WAR74] Warnier, J.,
Logical Construction of Programs.
H.E. Stenfert Kroese B.V., Leiden (1974).
- [WEG72] Wegner, P.,
The Vienna Definition Language.
ACM Computing Surveys, Vol. 4, No.1, (March 1972) pp.5-63.
- [YOS81] Yoshikawa, H.,
General Design Theory and a CAD System.
In T. Sata and E.A. Warman (Eds) Man-Machine Communication in CAD/CAM, North Holland, Amsterdam (1981) pp.35-53.
- [YOU75] Yourdon, E., L. Constantine,
Structured Design.
Yourdon Inc., New York, N.Y. (1975)

I. Függelék

Asszociatív adatbázisfelület bináris relációk kezelésére

A bináris relációkat az alábbi hármassal fejezzük ki:

$$\langle A, R, B \rangle$$

Jelentése: "A" "R" relációban van "B"-vel. Az adatbázis ilyen

hármassok halmaza: $DB = \{ \langle X, Y, Z \rangle \}$.

Ha $\langle A, R, B \rangle$ eleme DB-nek, akkor azt mondjuk, hogy $\langle A, R, B \rangle$ fönnáll.

DB minden $\{ \langle A_i, R, B_j \rangle \}$ részhalmaza (A_i, B_j tetszőleges, $R = \text{const.}$)

rendezett. Így bármely $\langle A_{i_0}, R, B_{j_0} \rangle$ elemnek van rákövetkezője, vagy

ő az utolsó, továbbá — ha a halmaz nem üres — első eleme is van a

halmaznak. A rendezettség természetes kialakulása a realizációban

pl. a relációhármassok létrehozási sorrendje. Egy-egy reláció mind

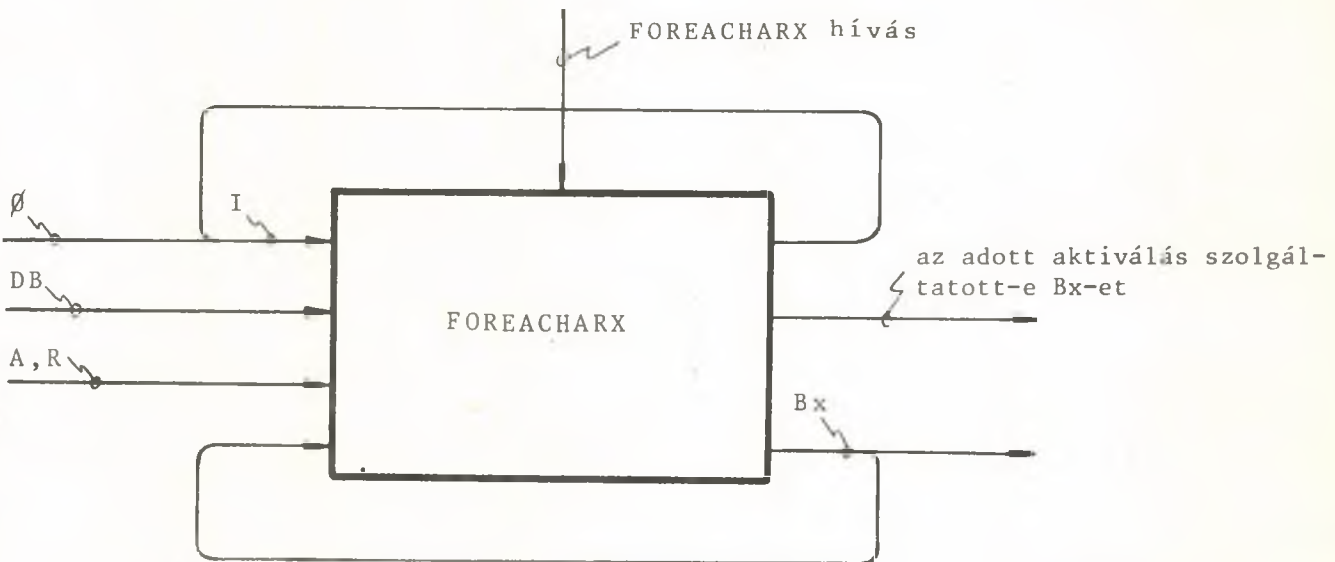
előtagja, mind utótagja szerint visszakereshető.

Az adatbázisfelület a következő funkciókat tartalmazza:

CREARB (A,R,B):	eredményként tárolja az $\langle A, R, B \rangle$ relációt DB-ben.
DEL(A,R,B):	törli $\langle A, R, B \rangle$ -t az adatbázisból.
ARB(A,R,B):	logikai függvény, eredménye igaz, ha $\langle A, R, B \rangle$ fönnáll hamis, ha nem áll fönn.

- ARX(A,R): függvény, eredménye egy olyan B, melyre $\langle A,R,B \rangle$ fönnáll.
- XRB(R,B): függvény, eredménye olyan A, melyre $\langle A,R,B \rangle$ fönnáll.
- FOREACHARX(A,R,B_x,I): függvény. Ciklikus végrehajtásának eredményeként B_x végigfut {B_x} halmazon, melynek minden elemére $\langle A,R,B_x \rangle$ teljesül. A függvény értéke igaz, ha a végrehajtás eredményeképpen tudott B_x-nek új értéket adni.
- A függvény I állapotjelzőjét a hívó szolgáltatja, így több FOREACHARX ciklus egymásba ágyazható (a FOREACHXRB ciklusokkal vegyesen – ld. alább).
- FOREACHXRB(A_x,R,B,I): függvény, mely az előbbihez hasonlóan, de a reláció előtaghalmazán fut végig.
- Mindkét FOREACH ciklus a FOREACHxxx függvénybe való ismételt belépéskor megvizsgálja, hogy az előzőleg átadott elem (B_x ill. A_x) által képzett reláció még eleme-e az adatbázisnak. Ha igen, akkor a rendezett $\{\langle A,R,B_x \rangle\}$ ill. $\{\langle A_x,R,B \rangle\}$ halmaz következő elemét szolgáltatja. Ha nem, akkor a halmaz első eleménél kezdi újra.

Az adatbázisfelület a relációhármassok tartalmáról csak annyit tud, hogy A és R tetszőleges n bites szám, R pedig n-1 bites szám. (A jelenlegi realizációban n=16.) Az n bites szám értelmezése teljes egészében a relációs felületet használó program feladata.

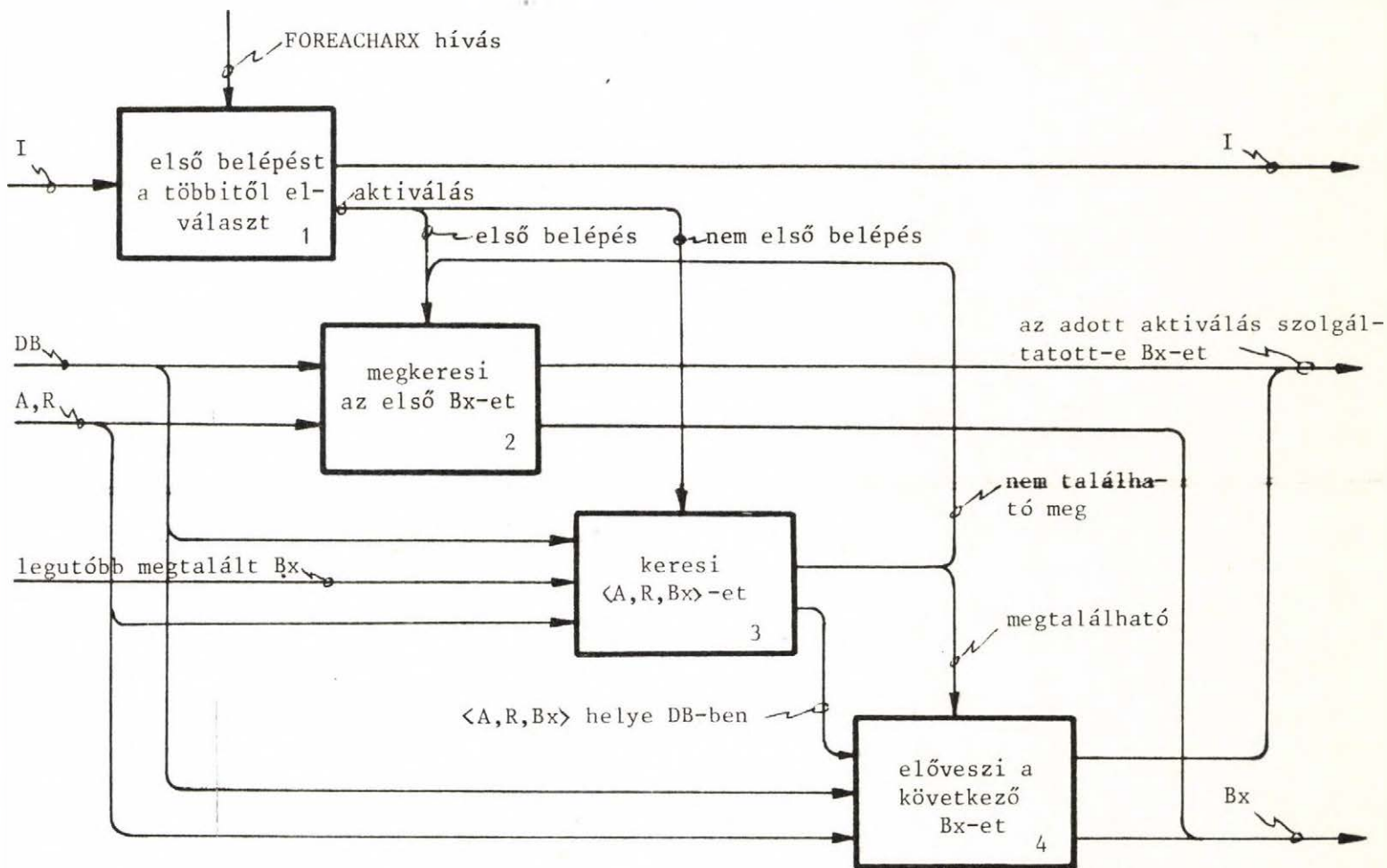


megjegyzés:

- I: a funkció állapotjelzője
ismert reláció-előtagok
Bx: Az ismeretlen {Bx} halmaz azon eleme, melyet a funkció hívásakor szolgáltat
DB: az adatbázis teljes tartalma
(megvalósítása egy AR és RB szerinti hash-tábla)

Tervezte	Bernus	Cím	FOREACHARX ciklikus vissza- keresés tartalom szerint				Cég
Rajzolta							A projekt megnevezése
Gépelte							
Ellenőrizte							
Kettesítés		Munkapéldány		Oldalszám	A szülő változatszám	Változatszám	
		Aktuális példány			A szülő változatszám	Változatszám	
Az elavulás időpontja		Jövőhagyva			Diagramkód	A-0	

Tervezte	Bernus	Cím	Cég
Rajzolta		FOREACHARX ciklikus vissza- keresés tartalom szerint	A projekt megnevezése
Gépette			
Ellenőrizte			
Keltezés		Munkapéntdőny	Változátszám
Az elavulás időpontja		Ártudás példány	Változás szám
		Jóváhagyva	Diagram kód
			AO



megjegyzés: A1 akkor aktiválja A2-t, ha I=0, egyébként A3-t indítja.
A1 I=1 értéket ad.

II. Függelék

SADT diagrammok analízise SDLA segítségével

(példa)

A SADT modellek diagrammokból állnak. Minden diagramm egy funkció (F), vagy adat (D) funkcionális definícióját adja. Közöttük egy funkcionális modellen belül az alábbi relációk értelmezhetők:

<u>része</u> (D,D)	<u>leszármazottja</u> (D,D)
<u>része</u> (F,F)	<u>leszármazottja</u> (F,F)
<u>input</u> (D,F)	
<u>output</u> (F,D)	
<u>kontroll</u> (D,F)	
<u>struktúra-mechanizmus</u> (F,F)	
<u>megvalósító mechanizmus</u> (D,F)	

Egy aktigrammon a definiálni kívánt funkciót a rétegnek megfelelő nézőpont szerint a része (F,F) reláció szerint perfekt diszjunkt módon bontjuk.

Az így definiált újabb funkcióhoz input, output, kontroll és mechanizmus reláción keresztül kapcsolódhatnak dolgok.

Az input olyan adatosztály, melyet (vagy melynek egy részét) a funkció (vagy annak egy része) használ. A kontroll és a mechanizmus speciális szerepű input. Az output olyan adatosztály, melyet a funkció, vagy annak legalább egy része létrehoz.

Az aktigramm tehát formailag egy diagramm, amely nyilakból és dobozokból áll. Minden doboz megfelel egy részfunkciónak, minden nyíl egy adatosztálynak.

A részek kapcsolatát leíró nyílhálózat elágazása leszármazottja relációt generál a szögpontba bemenő és abból kimenő élek reprezentálta adatosztályok között.

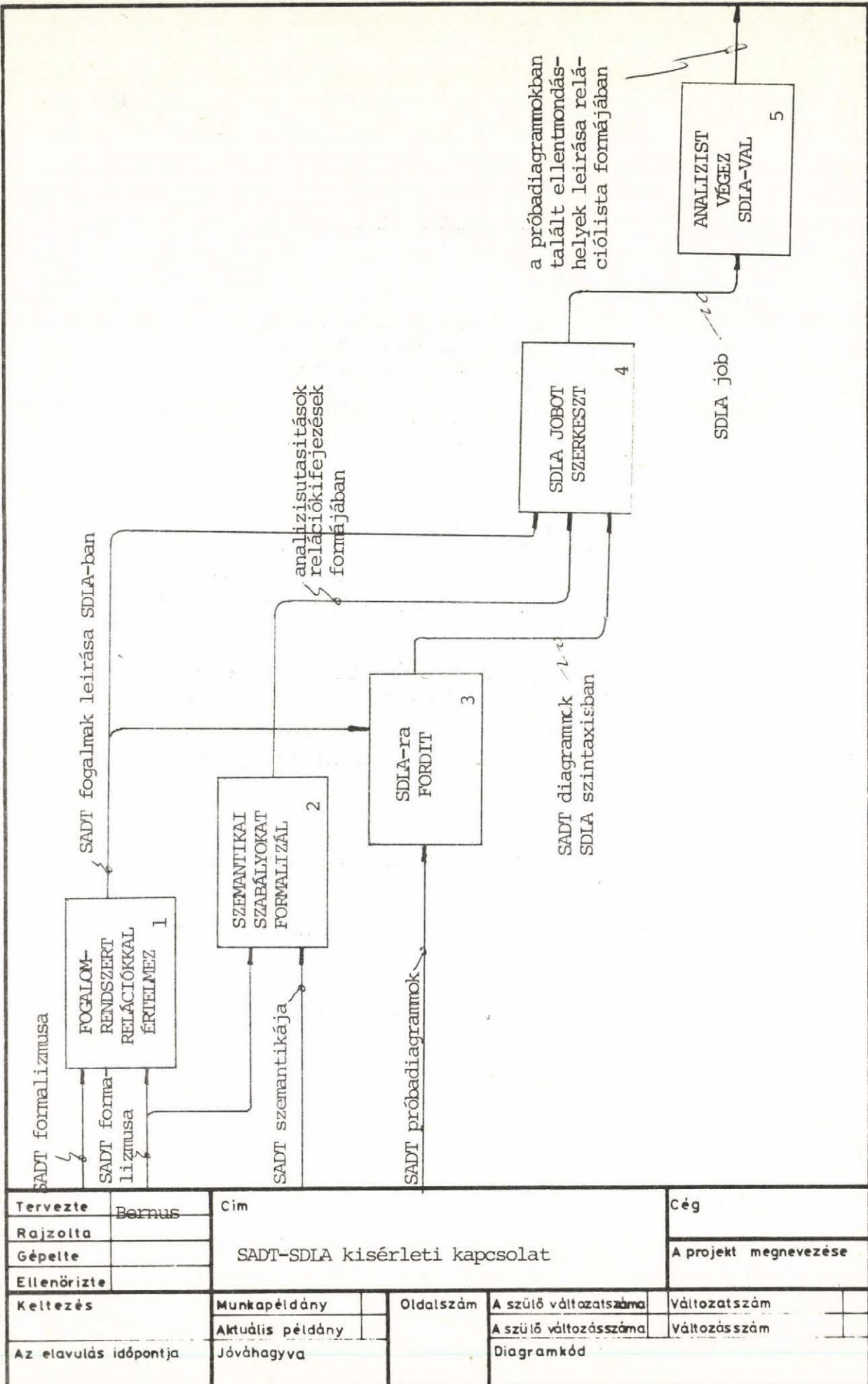
Jelöljük egy funkcionális architektúra *i*-edik modelljéből következő leszármazottja relációk halmazát

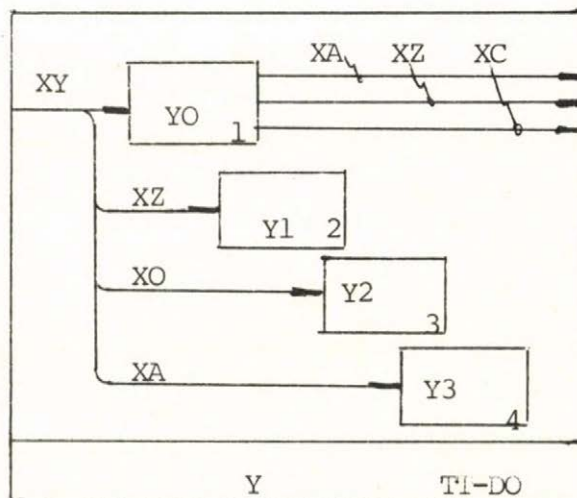
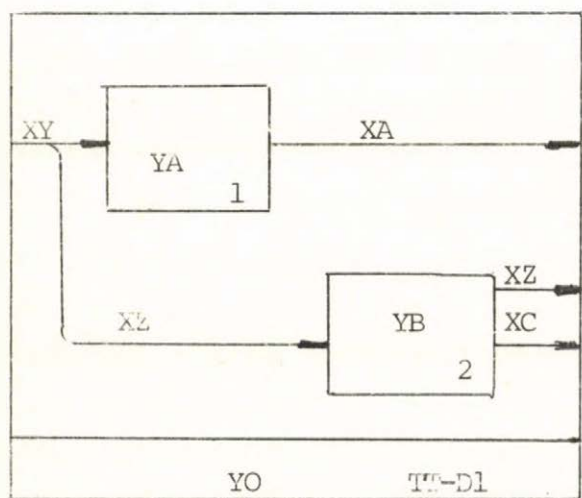
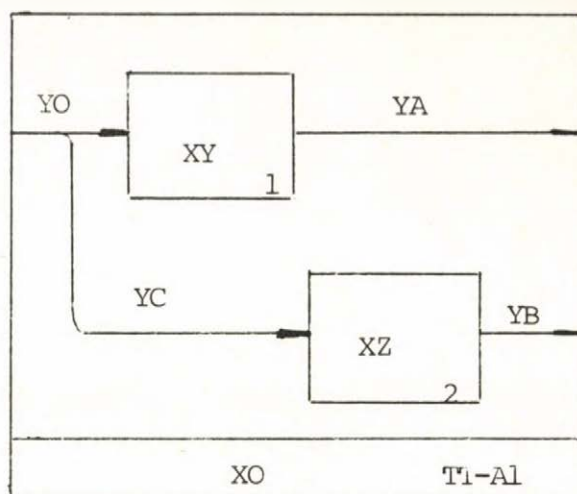
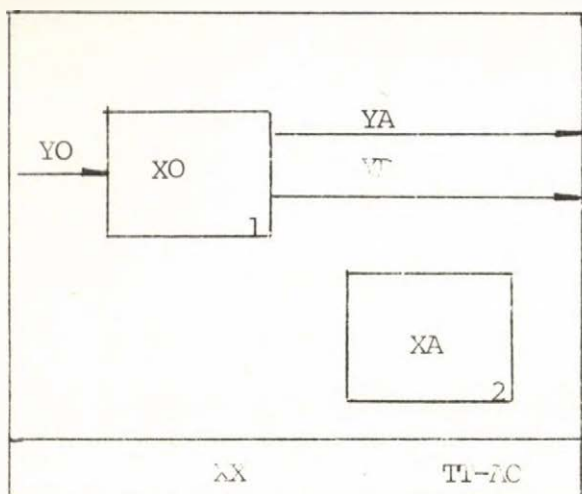
l_i -vel, ha aktigrammról,
 L_i -vel, ha datagrammról

származik. Jelöljük továbbá a része relációk halmazát

r_i -vel, ha datagrammról,
 R_i -vel, ha aktigrammról

származik. Legyen ezekívül F az irányított gráfokra értelmezett fedés





SADT próbadiagrammok

- II. függelék 2. ábra -

operátor. Ekkor az i -edik modell aktigrammjai és datagrammjai közötti ellentmondásmentesség feltétele, hogy

$$(l_i + L_i) \setminus F(r_i + R_i) = \Phi$$

üres halmaz legyen. Az analízis eredményének értékelhetősége szempontjából hasznos, ha az l_i és L_i relációkhoz harmadik elemként hozzávesszük azon aktigramm illetve datagramm kódját, melyen indukálódott. Ezen esetben Φ elemei nemcsak az ellentmondás tényét, hanem helyét is mutatják.

A SADT modellek leírásához szükséges előbb említett relációk találhatóak a mellékelt számítógépes lista DEFUNIT részében.

A SUBPROCESS és SUBDATA relációk a része (F,F) és a része (D,D) relációknak felelnek meg. A DESCENDANTPROC és DESCENDANTDATA a leszármazottja (F,F) és a leszármazottja (D,D) relációk megfelelői. A USE az inputhoz, a CONTROL a kontroll-hoz, a DERIVATION az output fogalmakhoz kapcsolható.

Az indukálás helyének a relációhoz kapcsolására szolgálnak aDEFINITION fogalmak.

Az így kialakított fogalomrendszerrel egy példadiagrammsorozat ellentmondásmentességi vizsgálatát végeztük el. Az erre a célra készült, egy hibát tartalmazó és nem teljes diagrammsorozat a II. függelék 2.ábráján látható. Az SDLA leírás a csatolt számítógépi listán, a DATAUNIT részben olvasható.

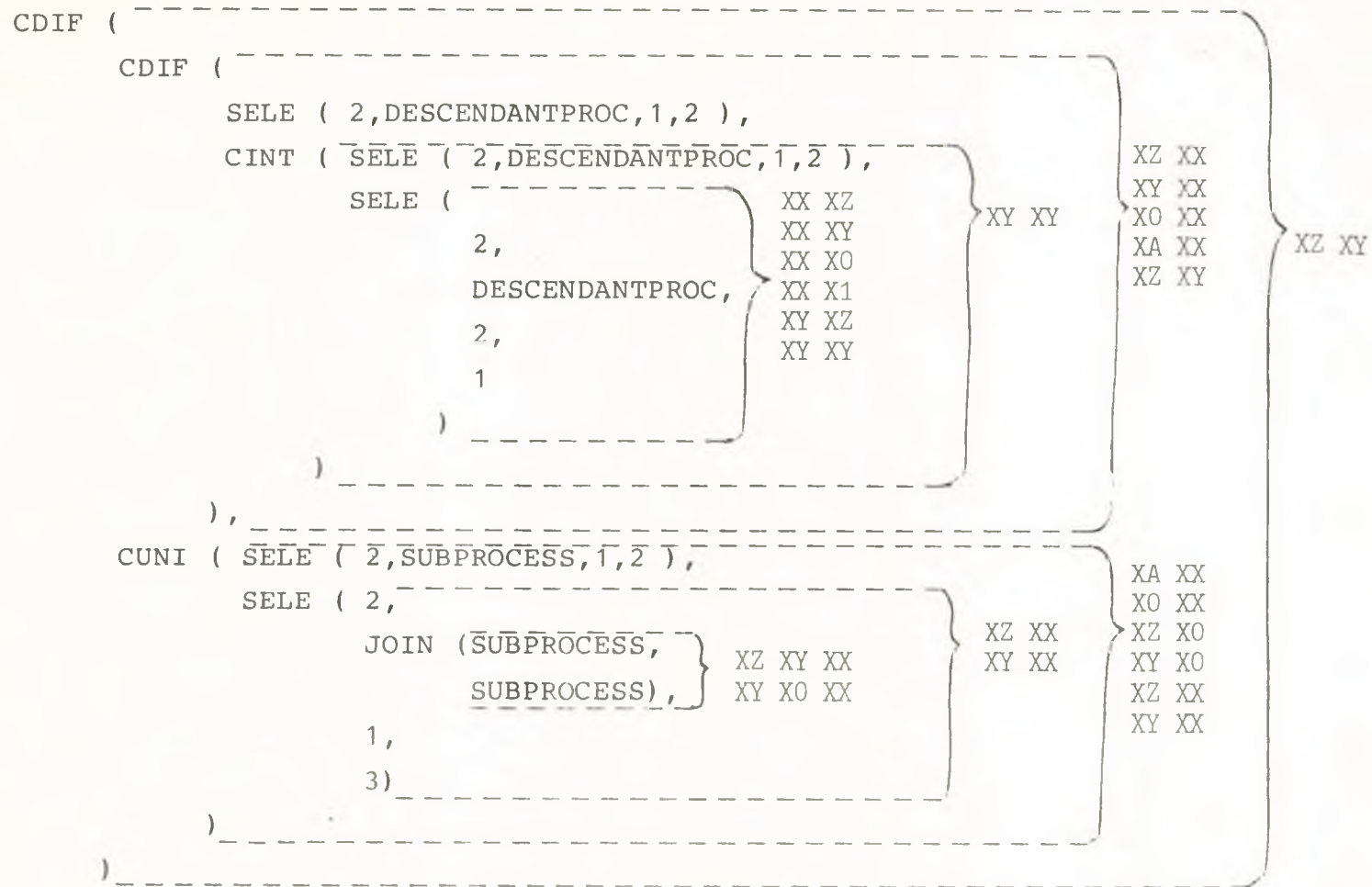
A vizsgálathoz szükséges tartalom szerinti relációműveletek részei az SDLA rendszer CDC 3300-as, SIMULA67-es implementációjának. (Ezek a CINT, CDIF, CUNI operátorok, amelyekkel metszetet, különbséget és úniót lehet képezni.)

A SELE (select) és a JOIN műveletek az eredeti értelemben használatosak.

A példaként választott analíziskifejezés részeredményeit a mellékelt ábrán feltüntettük. A teljes kifejezés az ellentmondásreláció Φ halmazát hozza létre.

A kifejezés első része kiszűri a tranzitivitásokat a leszármazottja relációkból (valamely D vagy F leszármazottja önmagának), második része pedig elkészíti a része relációval kompatibilis leszármazottja relációk halmazát. (A része reláció tranzitív lezárását.)

A módszer hátránya az, hogy a nagymélységű lebontások esetén a SUBPROCESS-ek helyére be kell írni rekurzíve a CUNI kifejezést (a lebontás legnagyobb mélységének megfelelően). Ez viszonylag hosszú kiértékelési ciklust eredményezhet. Megoldást a részeredmények tárolása (az SDLA procedúrális hozzáférése) jelenthet.



Ellentmondáshelyet leíró analíziskifejezés és kiértékelése
/ II. Függelék /

DEFUNIT :

CONCEPT COMMENT (COMMENTED:UNIV,COMMENT:TEXT);
FORM COMMENTED: COMMENT COMMENT;

CONCEPT DIAGRAM;

CONCEPT ACTIGRAM IS DIAGRAM;
FORM ABSOLUTE : ACTIGRAM :

CONCEPT DATAGRAM IS DIAGRAM;
FORM ABSOLUTE : DATAGRAM ;

CONCEPT PROCESS (CODE:ACTIGRAM);
FUNCTION ;
FORM ABSOLUTE : PROCESS ;
FORM ABSOLUTE : PROCESS-SYN: CODE;

CONCEPT PROCDEFINITION (DEFINED:PROCESS,ON:ACTIGRAM);
FUNCTION ;
FORM DEFINED: DEFINED-ON ON;

CONCEPT DATA (CODE:DATAGRAM);
FUNCTION ;
FORM ABSOLUTE : DATA :
FORM ABSOLUTE : DATA-SYN: CODE;

CONCEPT DATADEFINITION (DEFINED:DATA,ON:DATAGRAM);
FUNCTION ;
FORM DEFINED: DEFINED-ON ON;

CONCEPT SUBPROCESS (PART:PROCESS,OF:PROCESS);
FUNCTION ;
FORM PART: IS-PART-OF OF;
FORM OF: CONTAINS PART;

CONCEPT SUBPROCDEFINITION (DEFINITION:SUBPROCESS,ON:ACTIGRAM);
FUNCTION ;
FORM DEFINITION: DEFINED-ON ON;

CONCEPT DESCENDANTPROC (PART:PROCESS,FROM:PROCESS);
FUNCTION ;
FORM PART: DESCENDS-FROM FROM;
FORM FROM: IS-ANCESTOR-OF PART;

CONCEPT DESPCDEFINITION (A:DESCENDANTPROC,ON:DATAGRAM,PN:INTEGER);
FUNCTION ;
FORM A: DEFINED-ON ON SERIAL-NUMBER PN;

CONCEPT SUBDATA(PART:DATA,OF:DATA);
 FUNCTION ;
 FORM PART: IS-PART-OF OF;
 FORM OF: CONTAINS PART;

CONCEPT SUBDATADEFINITION(DEFINITION:SUBDATA,ON:DIAGRAM);
 FUNCTION ;
 FORM DEFINITION: DEFINED-ON ON;

CONCEPT DESCENDANTDATA(PART:DATA,FROM:DATA);
 FUNCTION ;
 FORM PART: DESCENDS-FROM FROM;
 FORM FROM: IS-ANCESTOR-OF PART;

CONCEPT DESDATADEFINITION(A:DESCENDANTDATA,ON:ACTIGRAM,PV:INTEGER);
 FUNCTION ;
 FORM A: DEFINED-ON PV SERIAL-NUMBER PV;

CONCEPT USE(USER:PROCESS,USED:DATA);
 FUNCTION ;
 FORM USER: USES USED;
 FORM USED: IS-USED-BY USER;

CONCEPT CONTROL IS USE;
 FORM USER: IS-CONTROLLED-BY USED;
 FORM USED: CONTROLS USER;

CONCEPT DERIVATION(DERIVATOR:PROCESS,DERIVED:DATA);
 FUNCTION ;
 FORM DERIVATOR: DERIVES DERIVED;
 FORM DERIVED: IS-DERIVED-BY DERIVATOR;

CONCEPT USEDEFINITION(DEFINITION:USE,ON:DIAGRAM);
 FUNCTION ;
 FORM DEFINITION: DEFINED-ON ON;

CONCEPT DERIVEDDEFINITION(DEFINITION:DERIVATION,ON:DIAGRAM);
 FUNCTION ;
 FORM DEFINITION: DEFINED-ON ON;

ENDUNIT ;

NO ERRORS-UNIT/STATEMENT ACCEPTED

REPORT FORMS

USE

USEDEFINITION = DEFINED-ON ON;

ABSOLUTE

DATAGRAM = DATAGRAM ;
 DATA = DATA-SYN: CODE;
 PROCESS = PROCESS ;
 ACTIGRAM = ACTIGRAM ;
 PROCESS = PROCESS-SYN: CODE;
 DATA = DATA ;

DESCENDANTDATA

DESDATADEFINITE= DEFINED-ON OF SERIAL-NUMBER PN;

UNIV

COMMENT = COMMENT COMMENT;

SUBPROCESS

SUBPROCDEFINITE= DEFINED-ON ON;

PROCESS

PROCDEFINITION= DEFINED-ON ON;
 CONTROL = IS-CONTROLLED-BY USED;
 SUBPROCESS = IS-PART-OF OF;
 SUBPROCESS = CONTAINS PART;
 DESCENDANTPROC= DESCENDS-FROM FROM;
 DERIVATION = DERIVES DERIVED;
 DESCENDANTPROC= IS-ANCESTOR-OF PART;
 USE = USES USED;

DESCENDANTPROC

DESPROCDEFINITE= DEFINED-ON ON SERIAL-NUMBER PN;

DERIVATION

DERIVDEFINITE= DEFINED-ON ON;

SUBDATA

SUBDATADEFINITE= DEFINED-ON ON;

DATA

DERIVATION = IS-DERIVED-BY DERIVATOR;
 DATADEFINITION= DEFINED-ON ON;
 SUBDATA = IS-PART-OF OF;
 SUBDATA = CONTAINS PART;
 DESCENDANTDATA= DESCENDS-FROM FROM;
 USE = IS-USED-BY USER;
 CONTROL = CONTROLS USEF;
 DESCENDANTDATA= IS-ANCESTOR-OF PART;

SOLA-37TESTVERSION

SYSTEM NAME=UNFROJ.3

PHASE=DEJCR_P11

DATAUNIT ;

TT-A0: ACTIGRAM ;

XX: PROCESS-SYN: TT-A0;
CONTAINS XA;
DEFINED-ON TT-A0;
CONTAINS X0;
DEFINED-ON TT-A0;

TT-A1: ACTIGRAM ;

X0: PROCESS-SYN: TT-A1;
CONTAINS XZ;
DEFINED-ON TT-A1;
CONTAINS XY;
DEFINED-ON TT-A1;
USES Y0;
DEFINED-ON TT-A1;
DERIVES YB;
DEFINED-ON TT-A1;
DERIVES YA;
DEFINED-ON TT-A1;

XY: PROCESS-SYN: TT-A11;
USES Y0;
DEFINED-ON TT-A1;
DERIVES YA;
DEFINED-ON TT-A1;

XZ: PROCESS-SYN: TT-A12;
USES YC;
DEFINED-ON TT-A1;
DERIVES YB;
DEFINED-ON TT-A1;

TT-D1: DATAGRAM ;

Y0: DATA-SYN: TT-D1;
CONTAINS YB;
DEFINED-ON TT-D1;
CONTAINS YA;
DEFINED-ON TT-D1;
IS-DERIVED-BY XY;
DEFINED-ON TT-D1;
IS-USED-BY XA;
DEFINED-ON TT-D1;

YA: DATA-SYN: TT-D11;
IS-DERIVED-BY XY;
DEFINED-ON TT-D1;
IS-USED-BY XA;
DEFINED-ON TT-D1;

YB: DATA-SFN: TT-012;
IS-DERIVED-BY XZ;
DEFINED-ON TT-01;
IS-USED-BY XB;
DEFINED-ON TT-01;
IS-USED-BY XC;
DEFINED-ON TT-01;

PROCESS XX:

IS-ANCESTOR-OF XZ;
DEFINED-ON TT-00 SERIAL-NUMBER 6;
IS-ANCESTOR-OF XY;
DEFINED-ON TT-00 SERIAL-NUMBER 6;
IS-ANCESTOR-OF X0;
DEFINED-ON TT-00 SERIAL-NUMBER 6;
IS-ANCESTOR-OF XA;
DEFINED-ON TT-00 SERIAL-NUMBER 6;

PROCESS XY:

IS-ANCESTOR-OF XZ;
DEFINED-ON TT-01 SERIAL-NUMBER 8;
IS-ANCESTOR-OF XY;
DEFINED-ON TT-01 SERIAL-NUMBER 8;

DATA Y0:

IS-ANCESTOR-OF Y0;
DEFINED-ON TT-A1 SERIAL-NUMBER 5;
IS-ANCESTOR-OF YC;
DEFINED-ON TT-A1 SERIAL-NUMBER 5;

ENDCLOSE ;

Köszönetnyilvánítás

Köszönetemet fejezem ki mentoromnak, Dr. Hatvany Józsefnek, aki elindított, majd szakmai tanácsaival éveken keresztül irányított kutatói pályámon.

Köszönettel tartozom Dr. Nemes Lászlónak, akinek főosztályán lehetőséget és támogatást kaptam az ebben a dolgozatban ismertetett kutatások végzésére.

Külön köszönetet nyilvánítok Kovács Vilmos munkatársnak, akivel 1979 óta végzek közös kutatómunkát. Értékes meglátásaival sok segítséget nyújtott mind az elméleti, mind a gyakorlati eredmények eléréséhez. Megköszönöm neki, hogy a II. Függelékhez szükséges anyagokat rendelkezésemre bocsájtotta.

Végül, de nem utolsósorban Dr. Gerencsér Piroskát és Vészi Ágnest illeti köszönet azért a lelkes kutatómunkáért, melyet a funkcionális tervezési módszertannal valós rendszereken végzett analízis során kifejtettek, s akik Kovács Vilmossal együtt szerzőtársaim voltak a "Komplex gépipari rendszerek funkcionális tervezési módszertana" c. kézikönyvek megírásában.

